PROGRAM DOKÜMÜ EKİ

16

- Basic Dili Editörü
- Reass
- Kutu Kutu
- Clip-Save
- Hotwheels
- Diskfaster
- Data Yaratıcı

COMMODORE
AYLIK BİLGİSAYAR DERGİSİ

Commodore'un Ücretsiz Ekidir

Commodore

Basic Dili Editörü

İkinci sayımızda yayınladığımız BASIC DİLİ EDİTÖRÜ programını gelen istekler üzerine yeniden yayınlıyoruz. Programı yazabilmeniz için kullanmanız gereken MAKİNE DİLİ EDİTÖRÜ'NÜ ve yeniden yayınladığımız BASIC DİLİ EDİTÖRÜ'NÜ diskette ya da kasette Teşvikiye'de bulunan Danışma Merkezi'mizden temin edebilirsiniz.

TUNA ERTEMALP

Artık BASIC programları dergiden yazarken hiç zorluk çekmeyeceksiniz ve yazdığınız satırların, dergideki satırların aynısı olduğundan emin olacaksınız. Ve en önemlisi; yazmanız bittiğinde RUN komutundan sonra programınız kesinlikle çalışacak, hiçbir SYNTAX ERROR veya OUT OF DATA gibi hata vermeden.

Bu nasıl mı olacak? Sizin yazacağınız tek şey eğer hâlâ yazmadıysanız, MDE'yi yazmak sonra da onun yardımı ile bu sayımızda yayınladığımız BDE'yi bilgisayara vermek. BDE'yi diskete veya kasete kaydettikten sonra, sıra dergiden gördüğünüz programları yazmaya gelecek. Rahat ve güvenli bir şekilde.

Bundan sonra BASIC program listelerini, teknik bir sorun çıkmadığı sürece, BDE formatında yayınlayacağız.

1) Bir kere bu program listelerinde, PRINT ve INPUT gibi emirlerin arkasından gelen konuşma işaretlerinin arasında görülen ve renk değiştirme, ekranı silme vaya cursor'u hareket ettirme gibi işlemleri simgeleyen negatif basılmış grafik işaretleri ortadan kalkıyor. Onların yerine, hangi tuşa basmanız gerektiği yazılıyor. Örneğin konuşma işaretinden sonra F1 tuşuna bastığınızda görünen grafik işareti yerine, (F1) yazısını göreceksiniz. Böylece, kağıt üzerinde koyu olmaları nedeniyle kolayca okunamayan kontrol hareketlerinin yazımı hızlı ve kolay olacak. Yanda, tüm, kontrol karakterlerinin BDE karşılığı verilmiştir: (örneğin [SPACE] yazısını gördüğünüzde, sadece boşluk tuşuna basacaksınız. Sakın [SPACE] yazmayın!

Ayrıca renk kodları (CTRL-1), (CTRL-2)...(CTRL-8) ile Commodore tuşu ile oluşturulanlar da (C=1), (C=2) (C=3) ile gösterilmektedir. Bunun yanısıra. (CTRL-A), (CTRL-B)...(CTRL-Z) ile gösterilen karakterleri de, CTRL tuşu ile belirtilen harfe beraber basarak elde edebilirsiniz. Klavyenin sol tarafındaki fonksiyon tuşları ise (F1)...(F8) şeklinde görülecektir. Eğer (11RIGHT) gibi bir yazıya rastlarsanız, bu, cursor'u sağa götüren tuşa 11 kez basacağınız anlamına gelecektir.

2) BASIC satırları içindeki komutlar arasında gereken yerlerde boşluklar bulunacak ve böylelikle okunaklılık artacaktır.

3) Harf tuşlarına SHIFT veya C=tuşu ile beraber bastığınızda ortaya çıkan grafik karakterleri yerine, hangi tuşa basmanız gerektiği yazılacaktır. Örneğin altı çizil bir A harfi, SHIFT ile A tuşuna basmanız gerektiğini belirtecektir. Üstü çizili A harfi ise. Commodore tuşu ile A harfine basmanızı gerektirecektir. Böylece, özellikle yanyana geldiklerinde ayırt edilemez olan grafik işaretlerini yorulmadan doğru olarak yazabileceksiniz.

4) Ve sonuncu özellik... Bu size satırı doğru girip girmediğinizi bildirecek bir kontrol toplamı. Dergideki programı yazmadan önce BDE'yi yükleyip RUN komutunu vereceksiniz, 1 saniye kadar sonra BDE size

READY ile karşılık verecektir. Artık programınızın ilk satırını yazabilirsiniz. O satırın sonunda [RETURN] tuşuna bastığınızda bilgisayar ekranın sol üst köşesine 0-255 arasında bir sayı çıkaracaktır. Ayrıca bilgisayar, önce iki nokta üstüste (:) sonra da bir boşluk basarak, sizin en fazla 3 basamaklı bir sayı vermenizi bekleyecektir. Burada, dergide her satırdan sonra (...) şeklinde göreceğiniz. Sayıyı yazın. Eğer o satırı doğru yazmışsanız dergideki kontrol toplamı ile ekranda beliren sayı aynı olacaktır ve o sayıyı yazıp [RETURN]'e bastığınızda, ince bir ses duyacaksınız. Artık o satırı doğru yazdığınızdan emin olarak, bir sonraki satırı yazabilirsiniz.

Ancak bir hata yapmışsanız, örneğin bir sayı unutmuşsanız veya satır numarası yanlışsa veya iki harfin yerini karıştırmışsanız dergideki toplam ile bilgisayarın hesaplayacağı sayı farklı olacaktır ve dergideki sayıyı verdiğinizde, gürültülü bir ses işiteceksiniz. Geri dönüp, o satırı düzeltmeniz gerekir. Ancak bu şart değil. Bir sonraki satıra da devam edebilirsiniz, ancak ne yazık ki, programınız hatalı olarak kalır.

Burada dört noktayı belirtmemiz lazım. Kontrol toplamının oluşmasında komutlar arasındaki boşlukların bir önemi yoktur. Dolayısıyla okunaklılığı artırmak için kullanılan boşlukları yazmayabilirsiniz.

Bir diğer önemli nokta da SIMON'S BASIC ile ilgili. Bu program ile BDE'yi birlikte kullanmak mümkün değil. Bu nedenle, SIMON'S BASIC ile yazılmış programlarda kontrol toplamı bulunmayacak. Ancak okunaklılığı artıran diğer özellikler (kontrol, renk ve grafik karakterleri gibi) kullanılacaktır.

Öte yandan, eğer kontrol toplamlarını kullanmak istemiyorsanız (yani kendinize çok güveniyorsanız), ya BDE'yi hiç yüklemeyin, ya da eğer yüklemiş ve çalıştırmışsanız POKE 1.55 komutunu verin. Tekrar çalıştırmak içinse POKE 1.53 komutu gerekli. Ancak çok dikkatli olun!! Eğer 53 ve 55 dışında bir sayı kullanırsanız. C-64'ünüz kolayca kitlenebilir ve o ana kadar yazdıklarınız da boşa gider.

Ve son olarak... Eğer BDE çalışır durumda iken LOAD işlemi yaparsanız, sizden bir satır yazmışsınız gibi kontrol toplamı istenecektir. Bilgisayarın bu isteğini duymamazlığa gelip sorusuna [RETURN] ile cevap verebilirsiniz. Hiçbir yanı etkisi olmayacaktır.

Şimdi sıra BDE'yi (ya da önce MDE'yi) yazmanıza geldi Kolay gelsin...□

```
 PROGRAM   : BDE                    0801 0968
-------------------------------------------

0801 : 0D 08 C2 07 9E 32 30 36   4C
0809 : 33 3A A2 00 00 00 A9 00   A9
0811 : 85 FE A2 01 BD 64 09 85   0C
0819 : FF A0 00 B1 FE 91 FE 88   28
0821 : D0 F9 E6 FF A5 FF DD 66   46
0829 : 09 D0 EE CA 10 E6 A9 74   77
0831 : A0 08 85 5F 84 60 A9 EC   EE
0839 : A0 08 85 5A 84 5B A9 D8   05
0841 : A0 E4 85 58 84 59 20 BF   D3
0849 : A3 A9 EC A0 08 85 5F 84   43
0851 : 60 A9 64 A0 09 85 5A 84   E2
0859 : 5B A9 37 A0 E4 85 58 84   50
0861 : 59 20 BF A3 A9 35 85 01   8B
0869 : A9 60 8D 31 A5 A9 E4 8D   22
0871 : 32 A5 60 A0 02 A9 00 AA   65
0879 : 85 FE B1 5F F0 28 C9 20   8D
0881 : D0 03 C8 D0 F5 85 FF 8A   C0
0889 : 29 07 AA F0 0E 48 A5 FF   B8
0891 : 18 2A 69 00 CA D0 F9 85   3F
0899 : FF 68 AA E8 A5 FF 18 65   19
08A1 : FE 85 FE 4C 6F E4 C0 04   D4
08A9 : 30 DB C6 D6 A5 D6 48 A2   CB
08B1 : 03 A9 20 9D 01 04 BD D4   15
08B9 : E4 20 D2 FF D0 0C 00 5C   88
08C1 : 48 20 C9 FF AA 68 90 01   BE
08C9 : 8A 60 CA 10 E4 A6 FE A9   0B
08D1 : 00 20 CD BD A9 3E 20 D2   BF
08D9 : FF 68 85 D6 20 6C E5 A9   99
08E1 : 8D 20 D2 FF 4C BF E3 09   97
08E9 : 3C 12 13 A9 3A 20 D2 FF   18
08F1 : A9 20 20 D2 FF A9 00 85   65
08F9 : FB A0 04 88 F0 1F 20 CF   7F
0901 : FF C9 0D F0 18 38 E9 30   92
0909 : 48 A2 0A A9 00 18 65 FB   A8
0911 : CA D0 FA 85 FB 68 65 FB   43
0919 : 85 FB 4C CF E3 A9 8D 20   AB
0921 : D2 FF A9 00 8D 05 D4 A9   05
0929 : F8 8D 06 D4 A9 0F 8D 18   7D
0931 : D4 A5 FB C5 FE F0 09 A9   7E
0939 : E8 A2 06 A0 81 4C 1A E4   B5
0941 : A9 8A A2 3A A0 21 8D 00   68
0949 : D4 8E 01 D4 8C 04 D4 A2   C1
0951 : 00 A0 00 EA C8 D0 FC E8   D8
0959 : D0 F9 8E 18 D4 8E 04 D4   48
0961 : 4C 80 A4 A0 E0 C0 00 E0   00
```

Reass

TUNA ERTEMALP

Onüçüncü sayımızda, profesyonel bir assembler programı yayınlamıştık: SUPER-ASS. Şimdi de aynı kalitede bir Reassembler veriyoruz! REASS.

Öncelikle bilmeyenler için, reassembler'in ne olduğunu anlatalım.

Bildiğiniz gibi, assembler ile, makina dilindeki programlarımızı rahatça yazıp değiştirebiliyorduk. Peki, daha önceden yazdığımız, belki bir makina dili monitörü programıyla, belki de DATA satırlarıyla hazırladığımız, binbir emekle çalıştırmayı başardığımız programları daha da geliştirmeye çalışırken ne olacak? Ya gereken yenilikleri dededen kalma eski yöntemlerle gireceğiz (monitör, DATA'lar vs.) veya tüm programı assembler kodu haline çevirip rahata kavuşacağız. Çünkü, bildiğiniz gibi, assembler kodunda yapacağınız her türlü değişikliği SUPER-ASS kolayca makina koduna çevirecektir.

Peki ama, elimizdeki makina dili programı assembler koduna nasıl çevireceğiz? Eğer bir reassembler'iniz yoksa, bu işi yapmanın tek yolu, programı satır satır kağıda geçirmek (printer'iniz yoksa yandınız), sonra programda label (SUPER-ASS'da bu tür terimleri açıklamıştık) konabilecek yerleri bulmak, tabelaları bulup çıkarmak ve sonra da oturup tek tek SUPER-ASS'ı kullanarak bilgisayara assembler kodu olarak girmektir. Bu işlem en iyimser tahminle 5-6 saat sürer. Tabii programınız fazla uzun olmadığı sürece. Hele bir de başkasının yazdığı bir makina dili programını değiştirmek ve bu arada da SUPER-ASS'dan yararlanmak istiyorsanız yandınız.

Yanmamayı ve tüm bu işlemleri 5-10 saniyede bilgisayarın halletmesini, bu arada da ayaklarınızı karşı koltuğa uzatıp rahat etmeyi istiyorsanız, derdinizin çaresi REASS demektir.

Bu program, COMMODORE 64'ün içinde herhangi iki adres arasında bulunan makina dili programlarını, gerek RAM'da gerekse ROM'da olsun, SUPER-ASS'ın kullanabileceği biçimde assembler koduna dönüştürür. Sizin tüm yapacağınız, assembler koduna dönüşecek programın başlangıç ve bitiş adreslerini REASS'a bildirmek. Geriye kalanı o haledecektir. Hatta, REASS'ı çağırırken kullanacağınız 0-225 arası bir sayı ile, assembler kodunuzun sahip olacağı birtakım özellikleri de ayarlayabilir, kendi damak zevkinize göre bir kod üretebilirsiniz. Artık teknik ayrıntılara girebiliriz.

REASS tamamen makina dilinde yazılmıştır ve $C000-$CB00 arasını kaplar. Çalışması sırasında SUPER-ASS'ın hafızada bulunup bulunmaması farketmez. Ayrıca REASS, demin de bahsettiğimiz gibi, üreteceği assembler kodunun bazı özelliklerini değiştirebilmeniz amacıyla üç yeni komut getirmektedir. Şimdi kullanma yönetimine geçelim.

REASS'ı kullanırken SUPER-ASS'ın hafızada bulunmasının avantajı, üreteceğiniz assembler kodunu derhal kullanabilme şansına sahip olmanız. Ancak dezavantajı da, boş hafıza miktarının azalması. Bunun sonucunda, üreteceğiniz assembler kodu ile makina dili programın kendisi beraberce hafızaya sığmayabilir. Bunun bir çözüm yolu, SUPER-ASS'ı doldurmamaktır. Diğer bir çözümünü ise yazımızın ilerleyen kısımlarında göreceğiz.

Önce REASS'ı MDE kullanarak yazın ve SAVE edin. Sıra kullanmaya gelince...

Önce makina dili programınızı bilgisayara yükleyin ,8,1 veya ,1,1 ile . Yalnız dikkat, bu programın $C000-$CB00 arasını boş bırakması lazım. Çünkü buraya REASS yerleşecek. Eğer program bu adresler arasındaysa, iki çözüm yolu var. Ya programınızı makina dili monitörü ile kaydırırsınız (11. sayıya bakınız) ya da BASIC programından bu işi yaparsınız. Bu işi yapabilecek bir

BASIC programı genel olarak şu görüntüye sahip olacaktır:

```
10 A1=başlangıç adresi
20 A2=bitiş adresi
30 A3=yeni başlangıç adresi
40 FOR I=0 TO A2-A1
50 POKE A3+1, PEEK (A1+I)
60 NEXT I
```

Örneğin $C000-$CFFF arasındaki bir programı $8000'den sonraya kopyalamak için A1=49152, A2=53247 A3=32768 olması gerekir. Bu tür çevirimlerin nasıl yapıldığını daha önceki sayılarımızda birçok kereler anlatmıştık. Örneğin Ayhan Kalaylıoğlu'nun vaya Tuna Ertemalp'in yazılarına bakabilirsiniz.

Tüm bu işlemlerden sonra NEW yapın. Şimde de REASS'ı yükleyin, tabii ki ,8,1 veya ,1,1 ile. Yükleme bitince yeniden NEW verin.

Şimdi sıra geldi REASS'a makina dili programımızın nerede olduğunu söylemeye. Diyelim ki programımız $9000-$9100 arasında. Kullanacağımız komut:

```
SYS 49152, $9000, $9100
```

olacaktır. Dikkat ettiyseniz, başlarına dolar işareti koymak şartıyla, onaltılık sistemdeki sayıları da kullanabiliyoruz. Ama isterseniz normal sayıları da kullanabilirsiniz. Bu durumda yukarıdaki komutu şu şekilde verebilirdik:

```
SYS 49152, 36864, 37120
```

Ama herhalde ilk biçim daha anlaşılır oluyor, ne dersiniz?

Bu komutun bir başka özelliği biraz önce gördüğünüz kaydırma işlemleri ile ilgili. Demin $C000-$CFFF arasındaki programı $8000'e kaydırmıştık. Bu programı reassembler'e kullandırmak için şu komutu veriyoruz:

```
SYS 49152, $C000, $D000, $8000
```

Bunun anlamı, "Benim programım aslında $C000-$D000 arasında olacaktı ama şu anda $8000'de duruyor." demektir. DİKKAT: Programımız $CFFF'de bittiği halde bitiş adresi olarak $D000 yazdık. Bu hep böyle yapılmalıdır. Yani bitiş adresi yerine programın son byte'ından sonra gelen adres yazılmalıdır. Bir önce verdiğimiz örnekte de, esas program $9000-$90FF adresleri arasında bulunuyor olmalıydı. Buna benzer bir olay, makina dili monitörü programında da karşımıza çıkıyordu.

Bu komutla REASS'a, çevireceğimiz programın hafızadaki yerini bildirmiş ve üç yeni komutu aktif hale getirmiş olduk. REASS'ın çevirme işlemini yapmasına birazdan geleceğiz.

Bildiğiniz gibi, assembler kodu kullanılırken gereken yerlere label konabiliyor. Biz de üretilecek assembler kodunun istediğimiz adreslerde bir label kullanılmasını isteyebiliriz. Bunun için kullanılacak komut şudur: ← P adres

Örneğin: P $C000 verdiğimizde, daha sonra üreteceğimiz assembler kodunun, $C000'a karşılık gelen satırının başında bir label olacaktır. Bunun gibi birkaç P komutu daha verilebilir.

Bir de tablo meselesi var. Makina dili programının bir kısmı program değil de, program tarafından kullanılan byte'lerden oluşabilir. Bunların 6510 makina dili komutlarına çevrilmesi anlamsız olur. REASS çalışmasını sürdürürken, bu tip bölgelere rastlarsa makina kodu yaratmak yerine ".BY" komutu ile başlayan satırlar yaratacaktır. Ancak tabloların içindeki byte'lar tesadüfen bir makina dili koduna karşılık gelebilir. Örneğin programın içinde bir yerde üç tane arka arkaya gelen $20 değeri varsa, bu aslında üç tane SPACE anlamına gelebilmektedir. Ancak bu değerler aynı zamanda JSR $2020 komutuna da karşılık gelmektedir. Üretilecek assembler kodunda da böyle görünecektir. Bunu önlemek için en iyi yol, eğer biliyorsanız, programın içindeki tabloların yerlerini üretime başlamadan önce REASS'a bildirmektir. Diyelim ki programımızın $C100-$C235 ve $CF00-$CFFF adresleri arasında tablolar var. Şu komutları kullanacaksınız:

```
← T $C100,$C235
← T $CF00,$CFFF
```

Artık REASS buralarda makina kodu aramaya çalışmayacaktır.

Artık sıra geldi esas çevirme işlemini yapmaya... Bunun için kullanılacak komut:

```
← E sayı
```

Burada vereceğiniz sayı, 0-255 arasında olacak, tıpkı yazının başlarında belirttiğimiz gibi. Bildiğimiz gibi, 0-255 arası sayılar 8 bit ile gösterilmekte ve bu bit'ler de 0-7 arası numaralandırılmaktadır. Buna göre düşünerek aşağıdaki açıklamaları okuyunuz. Her bir bit'in 1 veya 0 olması, üretilecek assembler kodunun görüntüsünü etkileyecektir.

BIT 0: Eğer değeri 1 ise, programınızdaki sıfırıncı sayfaya ($00-$FF adresleri) hitap eden tüm komutların label'ları üç harften oluşacaktır. Eğer 0 ise, bu label'ler beş karakterden oluşacaktır.

Örnek: L23 veya L0023

BIT 1: Eğer değeri 1 ise, programdaki her RTS, RTI, BRK ve JMP komutundan sonra assembler koduna boş bir satır eklenecektir. Eğer 0 ise, bu işlem yapılmayacaktır. Bunu yapmak, üretilecek programın okunurluğunu arttıracaktır. Eğer hafıza azlığı sorununuz varsa, yapmamanız gerekir.

BIT 2: Eğer değeri 1 ise, programdaki direkt adresleme moduna sahip her komutun yanına, kullanılan byte değerinin ASCII karşılığı da konacaktır. Eğer 0 ise, bu yapılmayacaktır.

Örnek: LDA $41; "A"

Eğer byte değerinin ASCII karşılığı okunabilir bir karakter değilse. nokta konulacaktır.

Örnek: LDX $00; " "

BIT 3: Eğer değeri 1 ise, programdaki tablo kısımlarının oluşturacağı assembler kodu satırlarının arasına bir boş satır eklenecektir. Eğer 0 ise, bu yapılmayacaktır.

BIT 4: Eğer değeri 1 ise, programdaki direkt adresleme moduna sahip komutların arkasına ASCII karşılıklarını eklemek geçerli olsa bile, bu işlem tablo kısımlarında yapılmayacaktır. Eğer 0 ise, tabloların arkasına da ASCII karşılıklar eklenecektir.

BIT 5: Eğer değeri 1 ise, programda label kullanılan yerlerde, tablo adresi belirten label'ların başına T harfi, programın dışında kalan bölgelere hitap eden adresleri belirten label'ların başına da E harfi getirilir. Eğer 0 ise, sadece L harfi konur.

BIT 6: Bu bit'i birazdan anlatacağız.

BIT 7: Eğer değeri 1 ise, REASS programın $A000-$BFFF veya $D000-$FFFF gibi ROM altında kalan adreslerde olduğunu anlar ve bu adreslerdeki ROM'da bulunan programı değil de, aynı adreslerdeki RAM'da bulunan sizin programınızı çevirir. Eğer 0 ise, ROM'daki programlar çevrilir. Böylece sistem rutinlerini de assembler koduna çevirebilirsiniz.

Gerçi MAKİNA DİLİNE MERHABA ve COMMODORE 64 İLE GRAFİK yazılarımızda sıkça değindik, ama gene de bu bit değerlerini nasıl 0-225 arası sayılara çevireceğimizi görelim.

Diyelim ki, sıfırıncı sayfa adreslemelerinin üçer karakter olmasını (BIT 0=1), kimi komutların arkasına bir boş satır eklenmesini (BIT 1=1), direkt mod komutlarının ASCII karşılıklarını görmeyi (BIT 2=1), tablo satırlarının arka arkaya gelmesini (BIT 3=0), tablolarda da ASCII karşılıklarını görmeyi (BIT 4=0), T ve E harflerini kullanmayı (BIT 5=1) ve ROM'un altındaki RAM'ı düşünmemeyi (BIT 7=0) istiyorsunuz. BIT 6'yı da 0 olarak varsayalım (Bu satırların yazarı, devamlı bu kombinasyonu kullanmaktadır). Sonuç:

2↑0+2↑1+2↑2+2↑5=39

Yani ← E39 verdiğimizde daha önce belirttiğimiz sınırlar arasındaki programımız ← P ve ← T komutlarına göre ve demin belirttiğimiz görünüşe sahip olarak assembler koduna çevrilecektir. REASS'ın çalışmasını, ekrandaki gittikçe uzayan bir çizgiden anlayabilirsiniz. REASS işini bitirince LIST Yapın. Eğer SUPER-ASS'ı daha önceden yüklemişseniz /E komutunu kullanmanız daha iyi olacaktır. Karşınıza üretilen assembler kodu çıkacaktır. EĞER LIST KULLANIYORSANIZ, SAKIN SATIRLARDA BİR DEĞİŞİKLİK YAPMAYA KALKIŞMAYIN. Önce üretilen kodu bir yerlere SAVE edin, sonra SUPER-ASS'ı yükleyip çalıştırın, ürettiğiniz programı yeniden yükleyip /E yardımı ile değiştirmeye başlayın. Artık gerisi size ait.

Baştan beri anlattığımız şeyler çok karmaşık görünse de, bir-iki kullanımdan sonra saniyelik otomatik işlemlere dönüşüyorlar.

DİKKAT: Yazımızın başından beri verdiğimiz komutları ayrı ayrı satırlarda değil de, aynı satırda yazmalısınız. Aksi takdirde, bir sürü SYNTAX ERROR yaratırsınız. Örneğin:

```
SYS 49152,$9000,$9100:
← P $9000: ← T$90F0,$90FF:
← E 39
```

Bu şekilde ancak 80 karakter (iki ekran satırı) uzunlukta komutlar yazabilirsiniz. Ancak, programınızın içinde 5-6 tane tablo varsa, 3-4 yeri label ile işaretlemek istiyorsanız, 80 karakter size yetmeyecektir. Ne yapacağız? Yapılacak şey minik bir program yazmak. Tıpkı BASIC program gibi. Örneğimize bir bakın:

```
10 ← P $9000
20 ← P $9040
30 ← P $905F
40 ← P $9065,$9080
50 ← P $90F0,$90FF
60 ← P $39
```

Bu programı yazıp,

```
SYS 49152,$9000,$9100:RUN
```

verdiğinizde, REASS önce makina kodunun yerini öğrenecek, sonra kısa programdaki bilgileri okuyacak, sonra bu kısa programı silerek yerine assembler kodunu üretmeye başlayacaktır. Nasıl, basit değil mi?..

Bu tür kısa program yazmadan direkt modu kullanmayı ancak tablo vaya ekstra label gibi bir derdiniz yoksa yapmanızı tavsiye ederiz. Örneğin:

```
SYS $8000,$8357: ← E39
```

Bir başka direkt mod kullanımı da, makina dili programınızın içindeki tabloların yerlerini bilmiyorsanız ve REASS'ın size yardımcı olmasını istiyorsanız olacaktır. Işte burada, demin es geçtiğimiz 6 numaralı bit'i kullanacağız. Eğer bu bit 1 ise, REASS assembler kodu üretmeyecek, makina kodunun üstünden geçerek, nerelerde tablolar olduğunu bulup bunları demin bizim yazdığımız türde bir program şeklinde ← T komutları ile belirleyecektir. Örneğin:

```
SYS $8000,$9000: ← E 46
```

komutu işledikten sonra LIST yapınca, şuna benzer bir program görebilirsiniz (tabii $8000-$9000 arasında bir program varsa):

```
100 ← T $8100,$8105
110 ← T $8107,$8200
```

İşte burada sizin biraz hayal gücünüzü çalıştırmanız gerekiyor. REASS, sadece makina kodu karşılığını bulamadığı kısımları tablo olarak niteleyecektir. Diyelim ki, $8106 adresindeki byte'ın değeri $60. Bu değer, makina dilindeki RTS komutunun koduyla da aynı olduğu için, REASS o adresi tabloya dahil etmeyecektir. Yukarıdaki örneği gözönüne alırsak, aslında tablonun $8100-$8200 adreslerini tamamen kapladığını varsayabiliriz.

REASS'ın ürettiği tablo listesinin yardımıyla gerçek tablo adreslerini ayarlayıp, en son satıra da E 39 gibi bir komut koyup (gerekiyorsa programın başına P komutlarını da ekleyerek) SYS $8000,$9000: RUN ile gayet güzel bir assembler kodu sahibi olabilirsiniz.

İsterseniz örnek olarak REASS'ın kendisini assembler koduna çevirelim. Önce bilgisayarı kapatıp açarak içini temizleyelim. Şimdi REASS'ı yerine yükleyin. Sonra aşağıdaki program ile onu $8000'den sonraya kaydıralım:

```
10  A1=49152
20  A2=51968
30  A3=32768
40  FOR I=0 TO A2-A1
50  POKE A3+I,PEEK(A1+I)
60  NEXT I
```

Şimde NEW verip şu programı yazın:

```
10 ← P$C000
20 ← T$C813,$CAFF
30 ← E 39
```

Şimdi de şu komutu verin:

```
SYS 49152,$C000,$CB00,$8000:
RUN
```

Sekiz saniye içinde assembler kodu üretilecektir. SAVE edin ve SUPER-ASS'ı yükleyip çalıştırın. Oluşturduğunuz assembler kodunu yükleyin ve üzerinde istediğinizi yapın. Şimdi gelelim bazı ayrıntılara... Eğer REASS'ı kullanırken SUPER-ASS da aktif durumda ise, yazacağınız minik programın içinde her satır numarasından sonra bir eksi işareti koymayı unutmayın.

Bazı programlarda, örneğin kendini devamlı değiştiren programlarda, kimi adresler devamlı değişebilir. Bu durumda REASS bu adresleri birer label ile değil de, gerçek adreslerini onaltılık sistemde yazarak çevirecektir. Bu tür satırların arkasına üç soru işareti eklenecektir. Böylece tanınmaları kolaylaşacaktır.

Makina dili programcıları bilirler, BIT ve LDA komutunun kullanıldığı bir hile vardır. BIT $00A9 komutu aynı zamanda .BY $2C LDA#$00 anlamına da gelmektedir. Bunun derinliklerine inmeyeceğiz. Ancak REASS, bu üç byte'ı .BY şeklinde yazıp, assembler kodu şeklini de noktalı virgülün arkasına açıklama olarak yerleştirecektir.

ÖNEMLİ: Eğer oluşacak assembler kodunun hafızada değil de disket üstünde oluşmasını istiyorsanız (bu yöntemi sadece disket sürücüsü olanlar kullanabilir), önce OPEN ve CMD komutları ile bir dosya açın sonra REASS'ı işletin. Örnek verelim:

Demin REASS'ın kendisini çevirirken yaptıklarımıza geri dönelim.

Tüm hazırlıkları bitirip sıra SYS 49152... komutuna geldiğinde şu satırı direkt modda verin:

```
OPEN 1,8,1,"dosya,P,W":
CMD1:SYS 49152,$C000
$CB00,$8000:GOTO 10
```

Assemler kodu, adını verdiğiniz dosyada yaratılacaktır. Normal LOAD ile daha sonra yükleyebilirsiniz. Bu yöntemle hafıza sıkışıklığı ortadan kalkar.

Assembler kodunun bu yöntemle üretimi bittikten sonra RUN-STOP/RESTORE tuşlarına basmanız gerekiyor. Eğer üretim sırasında bir hata çıkarsa bilgisayarınız kilitlenecektir. Dikkatli olun!..

Dikkat ettiyseniz, yukarıda RUN yerine GOTO kullandık. Çünkü RUN, açtığımız dosyayı kapatacaktı. GOTO bunu önleyecektir.

REASS çalışırken bazı hata mesajları verebilir. Bunların anlamları şu şekildedir:

SYNTAX EROR: Ya ← T,, ← P ve ← E komutlarından biri yanlış kullanıldı, ya da onaltılık sayılardan biri 4 basamaktan az verildi.

OUT OF MEMORY: Ya hafıza yetmedi, ya da program 2700'den fazla label kullanımı gerektiriyor.

ILLEGAL QUANTITY: Ya onaltılık sistemdeki sayının başında dolar işareti yok, ya tablo başlangıç adresi bitiş adresinden sonra görünüyor, ya birkaç tablo bölgesi üstüste görünüyor, ya da belirtilen adreslerden bir kısmı SYS ile verilen başlangıç ve bitiş adreslerinin dışında.

TYPE MISMATCH: Onaltılık sistemdeki bir sayıda 0-9 ve A-F arası karakterlerden başkası da var.

Daha önce ← P ile belirlenmiş adresler, tablo başı veya sonu olarak belirlenemezler. SYS komutunda verilen bitiş adresi de tablo sonu adresi olamaz. Zaten bildiğiniz gibi bu adres, aslında programa dahil değil.

İşte bu kadar. Artık 11 numaralı sayımızda verdiğimiz makina dili monitörü programı ve 13 numaralı sayımızda verdiğimiz assembler programı SUPER-ASS eşliğinde, REASS ile birlikte makina dili programcılığı yolunda ilerlemenize yarayacak her türlü silahınız var. Gerisi size kalmış...

Hoşçakalın...

64 ER dergisinden uyarlanmıştır.

```
 PROGRAM  : REASS                    C000 CB00
-------------------------------------------------

C000 : 20 4E C1 20 53 E4 AD 05  D9
C008 : C8 8D 02 03 AD 06 C8 8D  C1
C010 : 03 03 AD FF C7 8D 0A 03  17
C018 : AD 00 C8 8D 0B 03 20 53  99
C020 : C7 8D 09 E0 8C 08 E0 8D  B4
C028 : 01 E0 8C 00 E0 8D 03 E0  05
C030 : 8C 02 E0 48 98 48 20 53  F1
C038 : C7 8D 05 E0 8C 04 E0 20  D0
C040 : 79 00 F0 09 20 53 C7 8D  ED
C048 : 01 E0 8C 00 E0 68 85 14  6C
C050 : 68 85 15 A9 FF A2 02 9D  4E
C058 : 28 E0 CA 10 FA AD 02 C8  5C
C060 : 8D 08 03 AD 03 C8 8D 09  27
C068 : 03 20 D1 C0 20 D7 C0 4C  64
C070 : 1E C1 A9 00 85 0D 20 73  01
C078 : 00 B0 03 4C F3 BC 20 13  E6
C080 : B1 90 03 4C 28 AF C9 24  33
C088 : F0 03 4C 9A AE 20 A0 C0  50
C090 : 85 62 20 A0 C0 85 63 A2  6D
C098 : 90 38 20 49 BC 4C 73 00  72
C0A0 : 20 BF C0 20 B5 C0 0A 0A  72
C0A8 : 0A 0A 85 14 20 BF C0 20  DE
C0B0 : B5 C0 05 14 60 C9 3A 90  E8
C0B8 : 02 69 08 29 0F 38 60 20  0B
C0C0 : 73 00 90 12 C9 41 90 04  8B
C0C8 : C9 47 90 0A A2 16 6C 00  27
C0D0 : 03 78 A9 35 85 01 60 A0  43
C0D8 : 00 20 DF C0 90 DF C8 A5  6F
C0E0 : 14 D9 02 E0 C8 A5 15 F9  7F
C0E8 : 02 E0 60 20 73 00 20 56  DB
C0F0 : C7 A2 80 20 36 C1 20 53  C5
C0F8 : C7 20 D1 C0 20 3C C5 B0  B8
C100 : 1D 20 09 C1 A2 00 4C 36  70
C108 : C1 20 57 C6 A0 00 B1 59  0B
C110 : F0 14 10 F5 C8 A5 14 D1  7B
C118 : 59 C8 A5 15 F1 59 B0 06  9A
C120 : A9 37 85 01 58 60 A2 0F  18
C128 : 20 20 C1 6C 00 03 20 73  D6
C130 : 00 20 56 C7 A2 02 20 69  5C
C138 : C0 20 D1 C0 20 27 C5 20  27
C140 : CD C5 D0 E4 A1 59 30 D8  18
C148 : 20 20 C1 4C EA A7 A0 0B  F7
C150 : 20 D1 C0 B9 00 03 48 B9  6D
C158 : 10 E0 99 00 03 68 99 10  39
C160 : E0 88 10 EF 4C 20 C1 20  93
C168 : 4E C1 6C 02 03 20 CD C1  DE
C170 : 20 BC C1 B0 F8 A0 00 8C  22
C178 : AF 02 2C A8 02 30 03 20  36
C180 : 20 C1 B1 5D AA 20 D1 C0  0D
C188 : E6 5D D0 02 E6 5E E6 5F  4D
C190 : D0 02 E6 60 A5 5F 85 14  BB
C198 : 29 3F D0 11 20 20 C1 A5  0C
C1A0 : 9A C9 08 B0 05 A9 C0 20  18
C1A8 : D2 FF 20 D1 C0 A5 60 85  82
C1B0 : 15 20 D7 C0 90 06 A9 80  C4
C1B8 : 8D AF 02 60 A0 01 A5 5F  11
C1C0 : D1 FD C8 A5 60 F1 FD 90  25
C1C8 : 03 EE AF 02 60 18 A5 FD  C8
C1D0 : 69 03 85 FD 90 02 E6 FE  8F
C1D8 : 60 20 70 C1 8E A9 02 BD  56
C1E0 : FF C8 85 5C BD FF C9 8D  4E
C1E8 : B0 02 F0 17 B0 4F 30 4D  99
C1F0 : 29 0F 4A B0 1E 4A B0 0C  59
C1F8 : 20 70 C1 8E AA 02 A2 00  D8
C200 : 8E AB 02 60 20 70 C1 8E  9A
C208 : AA 02 B0 32 20 70 C1 8E  CF
C210 : AB 02 60 20 70 C1 8A 30  78
C218 : 0E 18 65 5F 8D AA 02 A5  F9
C220 : 60 69 00 8D AB 02 60 49  C5
C228 : FF 85 5B E6 5B 38 A5 5F  6A
C230 : E5 5B 8D AA 02 A5 60 E9  1E
C238 : 00 8D AB 02 60 18 A9 81  9A
C240 : 69 00 8D B0 02 A9 02 85  A3
C248 : 5C 60 20 73 00 F0 04 C9  76
C250 : 5F F0 03 4C E7 A7 20 73  95
C258 : 00 C9 50 D0 03 4C 2E C1  3A
C260 : C9 54 D0 03 4C EB C0 C9  A3
C268 : 45 F0 05 A2 0B 6C 00 03  D5
C270 : A2 00 8E AE 02 20 73 00  7B
C278 : F0 03 20 9E B7 8E A8 02  5C
C280 : 20 D1 C0 20 3E C7 20 27  AE
C288 : C5 20 BC C1 B0 1F 20 D9  FD
C290 : C1 AD B0 02 30 10 29 0F  DB
C298 : F0 0C C9 08 F0 08 A2 01  DE
C2A0 : 20 BF C5 20 CA C5 AD AF  06
C2A8 : 02 F0 E3 30 17 A0 00 B1  FB
C2B0 : FD 10 DB 20 70 C1 90 FB  FF
C2B8 : A0 00 B1 FD D0 F5 20 70  A2
C2C0 : C1 4C A6 C2 20 F5 C7 20  BA
C2C8 : 3E C7 20 D1 C7 A5 2B 85  8D
C2D0 : FB A5 2C 85 FC 20 75 C7  90
C2D8 : A0 00 B1 FD C9 FF F0 35  6F
C2E0 : 20 2F C7 20 D7 C0 90 27  A2
C2E8 : 20 1F C6 20 78 C7 A2 00  3E
C2F0 : 20 A4 C6 AD A8 02 29 20  49
```

```
C2F8 : F0 05 A9 45 20 34 C6 20  7D
C300 : B6 C6 A9 3D 20 34 C6 A9  3D
C308 : 24 20 B8 C6 20 70 C7 20  28
C310 : CD C1 4C D8 C2 20 3E C7  A2
C318 : 20 6D C7 20 1F C6 20 78  7E
C320 : C7 A2 01 20 A4 C6 A5 5F  53
C328 : 8D AA 02 A5 60 8D AB 02  65
C330 : A9 24 20 B8 C6 20 70 C7  C9
C338 : 20 6D C7 A0 00 B1 FD 30  FB
C340 : 10 20 2F C7 A0 00 20 DF  6F
C348 : C0 B0 06 20 CD C1 4C 3B  78
C350 : C3 20 BC C1 90 03 4C 91  00
C358 : C4 20 1F C6 20 D9 C1 AD  00
C360 : AE 02 D0 3A AD B0 02 C9  87
C368 : 81 D0 33 AD AF 02 D0 2E  7E
C370 : A8 A9 80 91 57 C8 8C AE  8A
C378 : 02 AD 08 E0 91 57 C8 AD  C1
C380 : 09 E0 91 57 20 6F C6 B1  45
C388 : FD 8D 09 E0 91 57 88 B1  04
C390 : FD 8D 08 E0 91 57 88 A9  BB
C398 : 00 91 57 20 62 C5 20 78  00
C3A0 : C7 AD B0 02 30 1C 29 0F  51
C3A8 : F0 69 C9 06 D0 12 AD AB  2C
C3B0 : 02 D0 0F A2 02 20 A4 C6  74
C3B8 : A9 83 4C 70 C4 4C 53 C4  C9
C3C0 : B0 FB 20 A2 C6 AD B0 02  6B
C3C8 : 30 F0 29 10 F0 05 A9 28  EB
C3D0 : 20 34 C6 20 BF C5 20 75  56
C3D8 : C5 20 B8 C6 AD B0 02 29  6F
C3E0 : 70 F0 33 C9 10 D0 08 A9  C9
C3E8 : 29 20 34 C6 4C 16 C4 C9  23
C3F0 : 40 90 11 F0 05 A9 29 20  5D
C3F8 : 34 C6 A9 2C A2 59 20 30  55
C400 : C6 4C 16 C4 A9 2C A2 58  42
C408 : 20 30 C6 AD B0 02 29 10  87
C410 : 4C E1 C3 20 A2 C6 A5 5C  F1
C418 : C9 37 D0 06 20 75 C7 20  7F
C420 : A2 C6 AD AF 02 30 2A 20  11
C428 : 70 C7 A5 5C C9 38 90 19  44
C430 : AD AE 02 D0 0A A5 5F 8D  35
C438 : 08 E0 A5 60 8D 09 E0 AD  26
C440 : A8 02 29 02 F0 03 20 6D  F6
C448 : C7 AD AF 02 D0 43 4C 59  1D
C450 : C3 30 58 20 A2 C6 A9 23  93
C458 : 20 34 C6 AD AA 02 8D A9  3E
C460 : 02 A2 71 AD A8 02 29 04  0D
C468 : F0 02 A2 81 8A 8D B0 02  0E
C470 : 29 03 85 5B AA BD A8 02  27
C478 : 9D 1F E0 CA D0 F7 20 EC  5D
C480 : C6 AD B0 02 C9 70 B0 8E  89
C488 : 20 75 C7 20 A2 C6 4C D3  92
C490 : C3 20 2F C7 20 1F C6 20  7E
C498 : 63 C6 A0 00 8C AE 02 B1  30
C4A0 : FD F0 70 30 2E 20 B6 C6  84
C4A8 : 4C 5C C3 A2 00 A9 FF 81  B8
C4B0 : 57 AD AE 02 F0 03 20 4A  06
C4B8 : C5 20 7D C7 20 33 C6 20  DD
C4C0 : 30 C6 20 20 C1 A5 B8 20  CC
C4C8 : C3 FF 20 B5 AB 6A 85 9D  A9
C4D0 : 4C AB E1 20 E6 C7 20 3D  16
C4D8 : C2 AD A8 02 29 10 F0 05  BC
C4E0 : A9 71 8D B0 02 A0 00 84  E9
C4E8 : 5B 20 70 C1 A4 5B 8A 99  2A
C4F0 : 20 E0 E6 5B B0 0A C0 07  12
C4F8 : D0 EF 20 16 C6 4C D6 C4  3E
C500 : A0 00 B1 FD F0 04 09 E0  E1
C508 : 91 FD 20 E6 C6 20 5C C7  EB
C510 : 4C 16 C4 20 E6 C7 20 70  AA
C518 : C1 8E A9 02 20 3D C2 4C  5B
C520 : 9E C3 20 3C C5 90 10 AD  AC
C528 : 08 C8 85 59 AD 09 C8 85  72
C530 : 5A 4C 37 C5 20 57 C6 20  4F
C538 : 3C C5 90 F8 A0 02 B1 59  2E
C540 : C5 15 D0 05 88 B1 59 C5  6C
C548 : 14 60 38 A5 57 E9 03 85  28
C550 : 57 B0 02 C6 58 A0 01 AD  43
C558 : 04 50 31 57 C8 AD 05 E0  E5
C560 : 91 57 38 A0 01 B1 57 E9  3E
C568 : 01 31 57 C8 B1 57 E9 00  3E
C570 : 91 57 4C 6F C6 20 22 C5  2F
C578 : D0 08 A0 00 B1 59 C9 FF  81
C580 : 90 07 A9 37 85 5C A9 24  0F
C588 : 60 AD A8 02 29 20 F0 1C  B9
C590 : 20 D7 C0 90 1A A9 45 20  22
C598 : 34 C6 4C AC C5 20 57 C6  20
C5A0 : B1 59 F0 13 C9 FF F0 04  05
C5A8 : C9 80 D0 F1 A9 4C 60 A0  E3
C5B0 : 00 B1 59 C9 80 D0 E9 A9  A2
C5B8 : 54 20 34 C6 4C AC C5 AD  3F
C5C0 : AA 02 85 14 AD AB 02 85  3A
C5C8 : 15 60 20 22 C5 F0 44 86  5C
C5D0 : 61 A5 59 85 5B A5 5A 85  62
C5D8 : 5C A0 02 B1 5B B6 61 99  5F
C5E0 : 61 00 8A 31 5B 88 10 F3  38
C5E8 : 18 A5 5B 69 03 85 5B 90  C2
C5F0 : 02 E6 5C E0 FF D0 E2 A0  EC
C5F8 : 01 A5 14 31 59 C8 A5 15  A0
C600 : 91 59 A5 5C CD 0C C8 90  B4
C608 : 07 A5 5B CD 0B C8 B0 7D  27
C610 : A2 00 60 A2 0E 60 20 E6  51
C618 : C6 20 5C C7 20 70 C7 18  D3
C620 : AD AC 02 69 0A 8D AC 02  35
C628 : 90 03 EE AD 02 AE AD 02  FB
C630 : 20 34 C6 8A 2C A8 02 70  5E
C638 : 66 A4 9A C0 08 B0 53 A0  44
C640 : 00 91 FB E6 FB A4 FB 84  C2
C648 : 2D D0 02 E6 FC A4 FC 84  2D
C650 : 2E C4 38 B0 38 90 48 18  5E
C658 : A5 59 69 03 85 59 90 02  CE
C660 : E6 5A 60 A0 02 B1 FD 91  68
C668 : 57 88 10 F9 4A B0 F3 2C  99
C670 : A8 02 50 0B 18 A5 57 69  6E
C678 : 03 85 57 90 02 E6 58 A5  2A
C680 : 58 CD 12 C8 90 DC A5 57  92
C688 : CD 11 C8 90 D5 A2 10 4C  6D
C690 : 28 C1 48 A8 20 20 C1 38  FB
C698 : 20 D2 FF 20 D1 C0 68 A0  2B
C6A0 : 00 60 A6 5C BD 4B C8 20  3F
C6A8 : 34 C6 BD 87 C8 20 34 C6  8B
C6B0 : BD C3 C8 4C 34 C6 A9 4C  C3
C6B8 : 20 34 C6 AD A8 02 29 01  9B
C6C0 : F0 05 AD AB 02 F0 06 AD  2F
C6C8 : AB 02 20 D0 C6 AD AA 02  1F
C6D0 : 48 4A 4A 4A 4A 20 DB C6  BC
C6D8 : 68 29 0F C9 0A 90 02 69  D2
C6E0 : 06 69 30 4C 34 C6 20 78  1B
C6E8 : C7 20 A2 C6 A2 00 F0 05  39
C6F0 : A9 2C 20 34 C6 A9 24 20  C8
C6F8 : 34 C6 BD 20 E0 20 D0 C6  E3
C700 : E8 E4 5B 90 EB AD B0 02  36
C708 : 10 33 20 75 C7 20 2B C7  22
C710 : A2 00 BD 20 E0 A8 29 7F  1D
C718 : C9 20 90 04 C9 60 90 02  7C
C720 : A0 2E 98 20 34 C6 E8 E4  E8
C728 : 5B 90 E7 A9 22 D0 4B A0  11
C730 : 00 20 34 C7 C8 B1 FD 99  8C
C738 : 13 00 99 A9 02 60 AD 08  D1
C740 : C8 85 FD AD 09 C8 85 FE  EB
C748 : A2 04 BD FF DF 95 5C CA  0D
C750 : D0 F8 60 20 FD AE 20 8A  A3
C758 : AD 4C F7 B7 AD A8 02 29  9B
C760 : 08 F0 DA 20 70 C7 20 1F  3F
C768 : C6 A9 3B D0 0D 20 66 C7  E7
C770 : A9 00 20 34 C6 20 69 C7  4B
C778 : A9 20 4C 34 C6 2C A8 02  3F
C780 : 8E A8 02 50 60 20 D1 C7  CB
C788 : 20 F5 C7 20 75 C7 20 1F  ED
C790 : C6 A9 5F 20 34 C6 B1 57  F6
C798 : C9 FF F0 30 C9 80 F0 0C  1F
C7A0 : A9 50 20 B5 C7 98 20 34  5A
C7A8 : C6 4C 8B C7 A9 54 20 B5  99
C7B0 : C7 A9 2C D0 ED 20 34 C6  AF
C7B8 : C8 B1 57 8D AA 02 C8 B1  22
C7C0 : 57 8D AB 02 A9 24 20 B8  B7
```

```
C7C8 : C6 4C 74 C6 A9 45 20 34   58
C7D0 : C6 A9 5A 8D AC 02 A2 00   18
C7D8 : 8E AD 02 AD 0E C8 85 57   5F
C7E0 : AD 0F C8 85 58 60 AD A8   88
C7E8 : 02 29 20 F0 05 A9 54 20   D4
C7F0 : 34 C6 4C B6 C6 A5 9A C9   09
C7F8 : 08 90 EA 4C 34 C6 2C 72   9C
C800 : C0 2C 4A C2 2C 67 C1 2C   1F
C808 : 28 E0 2C F0 F7 2C 00 F8   9C
C810 : 2C F0 FF 2E 2E 2E 20 48   DF
C818 : 59 50 52 41 2D 52 45 2C   23
C820 : 20 50 55 42 4C 49 53 48   F3
C828 : 45 44 20 42 59 20 36 34   B8
C830 : 27 45 52 2C 20 28 43 29   B7
C838 : 20 31 39 38 35 20 42 59   56
C840 : 20 4D 2E 20 57 45 48 4E   F4
C848 : 45 52 20 2E 2E 2E 41 41   60
C850 : 41 42 42 42 52 42 42 43   52
C858 : 43 53 43 43 45 49 49 50   E2
C860 : 4C 44 42 44 43 4E 43 44   24
C868 : 4C 53 54 42 54 4F 53 42   4D
C870 : 49 50 4A 4C 4C 50 50 53   2D
C878 : 42 43 53 52 53 53 54 54   45
C880 : 54 54 3F 42 4A 52 52 45   22
C888 : 42 42 44 4E 53 43 43 45   AD
C890 : 4F 50 56 4C 4C 45 4D 50   EB
C898 : 4F 4E 4E 4C 44 45 56 45   7E
C8A0 : 4C 4F 50 45 44 42 59 4D   A7
C8A8 : 41 52 54 49 4E 4C 53 44   6E
C8B0 : 53 48 48 45 4E 4C 45 4F   D0
C8B8 : 54 54 41 53 58 58 3F 52   DB
C8C0 : 4D 54 54 51 41 59 43 44   EB
C8C8 : 4C 43 53 51 52 4C 53 43   10
C8D0 : 49 43 50 58 52 43 53 50   1F
C8D8 : 41 43 43 58 44 50 59 59   76
C8E0 : 58 43 41 49 58 41 59 54   F1
C8E8 : 58 41 52 59 52 41 50 44   9A
C8F0 : 45 56 49 4C 41 58 59 58   23
C8F8 : 41 53 3F 4B 50 49 53 38   23
C900 : 22 02 02 02 22 05 02 2B   8D
C908 : 22 05 02 02 22 05 02 0A   D4
C910 : 22 02 02 02 22 05 02 0C   5F
C918 : 22 02 02 02 22 05 02 27   3D
C920 : 04 02 02 24 04 30 02 14   1C
C928 : 04 30 02 24 04 30 02 20   54
C930 : 04 02 02 02 04 30 02 0E   DC
C938 : 04 02 02 02 04 30 02 3A   3C
C940 : 11 02 02 02 11 29 02 2A   CA
C948 : 11 29 02 39 11 29 02 17   26
C950 : 11 02 02 02 11 29 02 0D   A0
C958 : 11 02 02 02 11 29 02 3B   04
C960 : 03 02 02 02 03 09 02 26   F2
```

```
C968 : 03 09 02 39 03 09 02 0B   2E
C970 : 03 02 02 02 03 09 02 2F   14
C978 : 03 02 02 02 03 09 02 02   C2
C980 : 31 02 02 23 31 32 02 1C   7C
C988 : 02 35 02 23 31 32 02 06   C3
C990 : 31 02 02 23 31 32 02 1F   92
C998 : 31 36 02 02 31 02 02 28   21
C9A0 : 15 1D 02 28 15 1D 02 33   72
C9A8 : 15 21 02 28 15 1D 02 07   24
C9B0 : 15 02 02 28 15 1D 02 2E   EB
C9B8 : 15 34 02 28 15 1D 02 1B   E5
C9C0 : 0F 02 02 1B 0F 16 02 13   84
C9C8 : 0F 18 02 1B 0F 16 02 2D   CB
C9D0 : 0F 02 02 02 0F 16 02 19   7D
C9D8 : 0F 02 02 02 0F 16 02 10   73
C9E0 : 1E 02 02 10 1E 12 02 25   47
C9E8 : 1E 1A 02 10 1E 12 02 08   21
C9F0 : 1E 02 02 02 1E 12 02 2C   A3
C9F8 : 1E 02 02 02 1E 12 02 00   53
CA00 : 34 80 80 80 04 04 80 00   07
CA08 : 08 00 80 80 06 06 80 01   D5
CA10 : 54 80 80 80 24 24 80 00   3A
CA18 : 42 80 80 80 26 26 80 02   64
CA20 : 34 80 80 04 04 04 80 00   97
CA28 : 08 00 80 06 06 06 80 01   A6
CA30 : 54 80 80 80 24 24 80 00   5A
CA38 : 42 80 80 80 26 26 80 00   80
CA40 : 34 80 80 80 04 04 80 00   47
CA48 : 08 00 80 02 06 06 80 01   45
CA50 : 54 80 80 80 24 24 80 00   7A
CA58 : 42 80 80 80 26 26 80 00   A0
CA60 : 34 80 80 80 04 04 80 00   67
CA68 : 08 00 80 12 06 06 80 01   67
CA70 : 54 80 80 80 24 24 80 00   9A
CA78 : 42 80 80 80 26 26 80 80   C1
CA80 : 34 80 80 04 04 04 80 00   F7
CA88 : 80 00 80 06 06 06 80 01   7E
CA90 : 54 80 80 24 24 44 80 00   2F
CA98 : 42 00 80 80 26 80 80 08   83
CAA0 : 34 08 80 04 04 04 80 00   DB
CAA8 : 08 00 80 06 06 06 80 01   26
CAB0 : 54 80 80 24 24 44 80 00   4F
CAB8 : 42 00 80 26 26 46 80 08   86
CAC0 : 34 80 80 04 04 04 80 00   37
CAC8 : 08 00 80 06 06 06 80 01   46
CAD0 : 54 80 80 80 24 24 80 00   FA
CAD8 : 42 80 80 80 26 26 80 08   30
CAE0 : 34 80 80 04 04 04 80 00   57
CAE8 : 08 00 80 06 06 06 80 01   66
CAF0 : 54 80 80 80 24 24 80 00   1A
CAF8 : 42 80 80 80 26 26 80 00   40
```

Kutu Kutu

14. sayıda yayınlanan bu program elimizde olmayan nedenlerle eksik yayınlanmıştır. Özür dileyerek tekrar yayınlıyoruz.

TUNA ERTEMALP

Hani şu okul sıralarında gizli gizli oynadığımız oyun var ya... Evde antrenman yapmaya ne dersiniz?

Hiç sanmıyorum ama, gene de bilmeyenler olabilir diyerek, biraz oyunu anlatalım. Karelere ayrılmış 10*10'luk bir alan var. Daha doğrusu kareleri siz oluşturacaksınız. Sizin önünüze, karelerin köşeleri gelecek. Aralarını doldurmak size düşüyor.

Önce bir oyuncu oyuna başlar. Karelerden birinin bir kenarını, yan yana duran iki nokta arasına çizgi çizerek oluşturur. Aynı şeyi öbür oyuncu da yapar. Eğer çizilen çizgi ile, daha önce üç kenarı tamamlanmış bir karenin dördüncü kenarı kapatılıp kare tamamlanıyorsa, o karenin içi dördüncü çizgiyi çizen oyuncuya ait olur. Böylece bir kare elden oyuncunun bir çizgi çizme hakkı daha doğar. Eğer bu sefer de kare tamamlarsa, bir çizgi daha çizer ve bu böylece sürüp gider; ta ki tamamlayacak kare kalmayıncaya kadar. Artık sıra diğer oyuncuya gelmiştir.

Oyunu, iyi bir strateji kuran, birer birer kareler peşinde koşmak yerine kare dizileri ele gelmeye çalışan ve en önemlisi, rakibine en az kareyi yerip onun kendisine mümkün olduğu kadar fazla kare vermesini sağlayan oyuncu kazanır.

Programımız da aynı bu şekilde oynar. Çizgi koyacağınız yeri belirlemek için kullanacağınız cursor'u ister joystick-1 ile, isterseniz de CRSR tuşları ile oynayabilirsiniz. Çizgi çizmek istediğiniz yere gelince, ya ateş tuşuna ya da RETURN tuşuna basarak çizgiyi yerleştirebilirsiniz.

Bilgisayarın oynaması, 0-10 arası zorluk derecesinde olacaktır. 0 iken, bilgisayarınız hiç düşünmeyecektir. Çizgileri rastgele koyacaktır. Burada kazanmanız kolay. Ancak 10 yaptığınızda, karşınızda hiç de kolay olmayan bir rakip bulacaksınız.

Oyunun o anki durumunu hem renkli karelerle oyun sahasında izleyecek, hem de skor halinde ekranın altında göreceksiniz. Dua edin de, sizin skorunuz artsın, etraf sizin renginizle dolsun...

Biraz da satır satır inceleme yapalım:

10-15: Ekran renkleri ayarlanıyor ve joystick ile klavye arasında tercih yapmanız isteniyor.

20-25: J veya K tuşlarından biri kabul edilip, buna göre J/K değişkeni hazırlanıyor. Daha sonra bu değişken 70 numaralı satırda kullanılacak.

30-35: Zorluk derecesi sorulup, verilen cevap kontrol ediliyor. Eğer 0-10 arasında ise, zorluk belirten göstergeler hazırlanıyor. Sonra da, ekran ve renk hafızasının başlangıç adresleri değişkenlere yerleştiriliyor.

40-45: Oyun alanı hazırlanıyor.

50-55: İleriki ekran işlemlerini kolaylaştıracak fonksiyonlar tanımlanıyor.

60: Oyun alanının ortasını gösteren değişkenler hazırlanıyor. Sonra o noktada bulunan renk ve karakter kodları okunuyor.

65: Cursor yakılıp rengi değiştiriliyor.

70-80: Klavye veya joystick okunuruyor. Eğer joystick okunuyorsa ve kol o anda oynamamışsa, tekrar kontrol ediliyor.

85: Cursor'un yeni koordinatları belirleniyor. Eğer oyun alanının dışına taşıyorsa, olduğu yerde kalıyor.

90: Eğer çizgi çekilmesi gerekiyorsa 105'e atanıyor.

95-100: Cursor'un yeri değiştirilip eski yerinde olması gereken renk ve karakter, tekrar yerine konuyor. Yeni yerde bulunan karakter ve renk saklanıyor.

105-155: Oyuncunun çizgi çizme isteği kontrol edilip, eğer geçerli ise yerine getiriliyor. Eğer bir kare tamamlanmışsa, oyuncunun tekrar oynaması sağlanıyor.

160.305: İşte burada, verdiğiniz zorluk derecesine göre bilgisayarın nereye çizgi koyacağı hesaplanıyor.

PROGRAM ADI : KUTU KUTU

```
00010 POKE 53281,12:POKE 53280,0:PRINT "<CTRL-1><RVSON><CLR><2D
      OWN>"SPC(14)"KUTU<2SPACE>KUTU"                              <096>
00015 PRINT "<2DOWN><15SPACE><CTRL-7>JOYSTICK<DOWN><5LEFT>OR<DO
      WN><7LEFT>KLAVYE (J/K)<SHIFT-SPACE>?";                      <233>
00020 GET A$:IF A$<>"J" AND A$<>"K" THEN 20                       <016>
00025 PRINT A$:JK=-1:IF A$="K" THEN JK=0                          <119>
00030 INPUT "<3DOWN><RIGHT>ZORLUK (0-10)";SK:IF SK<0 OR SK>10 T
      HEN 30                                                      <084>
00035 SK=(10-SK)/10:TS=200-200*SK:DT=TS+25:SC=1024:CO=13*4096+8
      *256                                                        <177>
00040 PRINT "<CLR>":FOR I=1 TO 10:PRINT "<CTRL-1> Q Q Q Q Q Q Q
       Q Q Q<DOWN>":NEXT                                          <172>
00045 YS=0:VS=0:PRINT " <CTRL-7>SEN:"YS"<6SPACE><RVSON>C64:"VS    <038>
00050 DEF FNBX(LC)=(PEEK(LC+40)<>32)+(PEEK(LC+1)<>32)+(PEEK(LC-
      40)<>32)+(PEEK(LC-1)<>32)                                   <098>
00055 DEF FNVH(LC)=LC<>2*INT(LC/2)                                <110>
00060 SL=SC+450:CL=CO+450:X=10:Y=11:CC=PEEK(SL):CR=PEEK(CL)       <219>
00065 POKE SL,160:POKE CL,7:F=0                                   <163>
00070 IF JK THEN 80                                               <148>
00075 GOSUB 405:GOTO 85                                           <191>
00080 GOSUB 370:IF PC=0 THEN 80                                   <153>
00085 X=X+J:Y=Y+K:IF X<1 OR X>19 OR Y<1 OR Y>19 THEN X=X-J:Y=Y-
      K:GOTO 70                                                   <214>
00090 IF PC=99 THEN 105                                           <240>
00095 POKE SL,CC:POKE CL,CR:SL=SL+PC:CL=CL+PC:CC=PEEK(SL):POKE
      SL,160                                                      <067>
00100 CR=PEEK(CL):POKE CL,7:GOTO 70                               <128>
00105 L=(PEEK(SL+1)=81)+(PEEK(SL-1)=81)+(PEEK(SL+40)=81)+(PEEK(
      SL-40)=81)                                                  <086>
00110 IF L+(CC=32)=-3 THEN 120                                    <219>
00115 I=128:GOSUB 325:GOTO 70                                     <206>
00120 WH=1:ML=SL:GOSUB 310:IF NOT FNVH(SL) THEN 145               <199>
00125 IF X>1 AND FNBX(SL-1)=-4 THEN BX=SL-1:GOSUB 330:F=-1        <164>
00130 IF X<19 AND FNBX(SL+1)=-4 THEN BX=SL+1:GOSUB 330:GOTO 60    <038>
00135 IF F THEN 60                                                <051>
00140 GOTO 160                                                    <164>
00145 IF Y>1 AND FNBX(SL-40)=-4 THEN BX=SL-40:GOSUB 330:F=-1      <056>
00150 IF Y<19 AND FNBX(SL+40)=-4 THEN BX=SL+40:GOSUB 330:GOTO 6
      0                                                           <012>
00155 IF F THEN 60                                                <071>
00160 WH=2:F=0:CN=0:IF RND(0)<SK THEN 195                         <018>
00165 FOR I=80 TO 720 STEP 80:FOR J=2 TO 18 STEP 2:K=SC+I+J:IF
      PEEK(K)=32 AND FNBX(K)=-3 THEN 175                          <109>
00170 NEXT J,I:GOTO 195                                           <186>
00175 I=K:IF PEEK(I-40)=32 THEN I=I-40:GOTO 270                   <087>
00180 IF PEEK(I+40)=32 THEN I=I+40:GOTO 270                       <040>
00185 IF PEEK(I-1)<>32 THEN I=I+1:GOTO 290                        <082>
00190 IF PEEK(I+1)<>32 THEN I=I-1:GOTO 290                        <025>
00195 I=INT(INT(RND(0)*20)*40+21*RND(0)+SC+41):CN=CN+1:IF PEEK(
      I)<>32 THEN 195                                             <150>
00200 IF NOT ((PEEK(I+1)=81 AND PEEK(I-1)=81) OR (PEEK(I+40)=81
       AND PEEK(I-40)=81)) THEN 195                               <030>
00205 IF SK>.6 OR CN>TS THEN 235                                  <111>
```

```
00210 IF FNVH(I) THEN 225                                                    <109>
00215 IF FNBX(I-40)=-2 OR FNBX(I+40)=-2 THEN 195                             <221>
00220 GOTO 270                                                               <012>
00225 IF FNBX(I-1)=-2 OR FNBX(I+1)=-2 THEN 195                               <027>
00230 GOTO 290                                                               <054>
00235 IF FNVH(I) THEN 255                                                    <158>
00240 IF SK>.6 OR CN>DT THEN 270                                             <171>
00245 IF FNBX(I+40)=-2 AND FNBX(I-40)=-2 THEN 195                            <153>
00250 GOTO 270                                                               <042>
00255 IF SK>.6 OR CN>DT THEN 290                                             <188>
00260 IF FNBX(I+1)=-2 AND FNBX(I-1)=-2 THEN 195                              <014>
00265 GOTO 290                                                               <091>
00270 ML=I:GOSUB 310:IF FNBX(ML-40)=-4 THEN BX=ML-40:GOSUB 330:
      F=-1                                                                   <182>
00275 IF FNBX(ML+40)=-4 THEN BX=ML+40:GOSUB 330:GOTO 160                     <202>
00280 IF F THEN 160                                                          <191>
00285 GOTO 60:IF NOT (PEEK(I-40)=81 AND PEEK(I+40)=81) THEN 195              <050>
00290 ML=I:GOSUB 310:IF FNBX(ML-1)=-4 THEN BX=ML-1:GOSUB 330:F=
      -1                                                                     <239>
00295 IF FNBX(ML+1)=-4 THEN BX=ML+1:GOSUB 330:GOTO 160                       <228>
00300 IF F THEN 160                                                          <211>
00305 GOTO 60                                                                <013>
00310 CL=CO+ML-SC:POKE ML,67:IF FNVH(ML) THEN POKE ML,93:                    <118>
00315 I=85:IF WH=2 THEN I=150                                                <049>
00320 FOR J=1 TO WH:POKE CL,0:GOSUB 325:POKE CL,1:FOR L=1 TO 20
      0:NEXT :POKE CL,0:NEXT                                                 <003>
00325 FOR K=1 TO 200:NEXT :RETURN                                            <058>
00330 YS=YS+1:J=3:I=200:CL=CO+BX-SC:IF WH=2 THEN J=2:I=150:YS=Y
      S-1:VS=VS+1                                                            <038>
00335 POKE BX,160:POKE CL,1                                                  <227>
00340 FOR L=1 TO 3:POKE CL,J:GOSUB 325:POKE CL,1:FOR K=1 TO 200
      :NEXT :I=I+18:POKE CL,J:NEXT                                           <029>
00345 PRINT "<HOME><21DOWN><CTRL-7> SEN:"YS"<6SPACE><RVSON>C64:
      "VS:IF YS+VS<81 THEN RETURN                                            <090>
00350 PRINT "<HOME><19DOWN>":IF YS>VS THEN 360                               <143>
00355 PRINT "<CTRL-1><6SPACE>KAYBETTINIZ<5SPACE>":GOTO 365                   <067>
00360 PRINT "<DOWN>KAZANDINIZ!"                                              <101>
00365 END                                                                    <113>
00370 J=0:K=0:PC=0                                                           <215>
00375 I=PEEK(56321):IF (I AND 1)=0 THEN PC=-40:K=-1:RETURN                   <238>
00380 IF (I AND 8)=0 THEN J=1:PC=1:RETURN                                    <070>
00385 IF (I AND 2)=0 THEN PC=40:K=1:RETURN                                   <019>
00390 IF (I AND 4)=0 THEN PC=-1:J=-1:RETURN                                  <081>
00395 IF (I AND 16)=0 THEN PC=99                                             <101>
00400 RETURN                                                                 <204>
00405 J=0:K=0:PC=0                                                           <250>
00410 GET A$:IF A$="" THEN 410                                               <183>
00415 IF A$="<UP>" THEN PC=-40:K=-1:RETURN                                   <070>
00420 IF A$="<RIGHT>" THEN J=1:PC=1:RETURN                                   <058>
00425 IF A$="<DOWN>" THEN K=1:PC=40:RETURN                                   <036>
00430 IF A$="<LEFT>" THEN J=-1:PC=-1:RETURN                                  <238>
00435 IF ASC(A$)=13 THEN PC=99                                               <107>
00440 RETURN                                                                 <244>
```

Clip-Save

ARTUN ALPER

Bildiğiniz gibi, BASIC programınızın tamamını diskete veya kasete kaydetmek istediğinizde SAVE komutunu kullanıyorsunuz. Peki, ya programın sadece bir kısmını SAVE etmek ister misiniz?

Diyelim ki, bir program yazdınız. Mesela 2000 satırlık. İçinden 100 satırlık bir altprogramı bir başka programda da kullanmak istiyorsunuz. Her tarafta bir sürü MERGE programı var. Ama programın istediğiniz bir bölümünü SAVE edebilen bir program yok. Garip, değil mi?

Artık var. Ama ne yazık ki, sadece disket kullanıcıları için. Şimdi CLIP-SAVE programını MDE ile yazın ve diskete kaydedin. Yüklerken, 8,1 kullanmayı unutmayın. Program $033C-$03CB adresleri arasına yerleşecektir. Şimdi de BASIC programını yükleyin. Diyelim ki, programınızın satır numaraları 100-900 arasında. Şimdi aşağıdaki örneklere bir bakın:

1) Sadece 500. satırı SAVE etmek için:

```
SYS 828,"isim",500
```

2) 500'den sonraki tüm satırları SAVE etmek için:

```
SYS 828,"isim",500-
```

3) 500-700 arasını SAVE etmek için:

```
SYS 828,"isim",500-700
```

4) 400'e kadar olan tüm satırları SAVE etmek için:

```
SYS 828,"isim",-400
```

Gördüğünüz tüm örneklerde programın belirttiğiniz kısmı "isim" adı altında diskete SAVE edilecektir. Satır numaraları verilişinin LIST komutu ile büyük benzerlik gösterdiğine dikkatinizi çekeriz.

Programımızın yaptığı şey, sizin verdiğiniz başlangıç ve bitiş satır numaralarına göre hafızanın hangi adresleri arasının SAVE edileceğini hesaplamak ve bu hafıza blokunu normal SAVE işlemi ile göndermek. Kaset kullanamayışımızın nedeni, CLIP-SAVE programının kaset için kullanılan tampon bölgede yeralması. Size ayrıca verdiğimiz SUPER-ASS formatındaki assembler dökümünü inceleyerek kendi isteğinize göre bir rutin yazabilirsiniz.

```
 PROGRAM  : CLIP-SAVE          033C 03CB
-----------------------------------

033C : 20 FD AE 20 57 E2 A9 08   4E
0344 : 85 B8 20 FD AE F0 7D 90   77
034C : 4B C9 AB D0 77 20 73 00   C7
0354 : 20 C2 03 A5 2B A6 2C 85   EE
035C : C1 86 C2 A0 05 B1 5F F0   62
0364 : 08 E6 5F D0 F8 E6 60 D0   BB
036C : F4 A6 60 A5 5F 69 07 90   FF
0374 : 01 E8 85 AE 86 AF A0 06   95
037C : A2 00 B1 5F 48 8A 91 5F   54
0384 : C8 B1 5F 48 8A 91 5F 20   F9
038C : FA F5 A0 07 68 91 5F 88   2B
0394 : 68 91 5F 60 20 C2 03 A5   18
```

```
033C : 5F A6 60 85 C1 86 C2 20   B3
03A4 : 79 00 F0 B7 C9 AB D0 1C   C6
03AC : 20 73 00 F0 06 20 C2 03   16
03B4 : 4C 5F 03 A5 2D A6 2E 85   F1
03BC : AE 86 AF 4C FA F5 20 6B   D9
03C4 : A9 4C 13 A6 4C 08 AF 00   F1

READY.
```

```
80   -;          CLIP-SAVE
90   -;
100  -           .EQ EL2B=$2B
110  -           .EQ EL2C=$2C
120  -           .EQ EL2D=$2D
```

```
130  -              .EQ EL2E=$2E
140  -              .EQ EL5F=$5F
150  -              .EQ EL60=$60
160  -              .EQ EL73=$73
170  -              .EQ EL79=$79
180  -              .EQ ELAE=$AE
190  -              .EQ ELAF=$AF
200  -              .EQ ELB8=$B8
210  -              .EQ ELC1=$C1
220  -              .EQ ELC2=$C2
230  -              .EQ ELA613=$A613
240  -              .EQ ELA96B=$A96B
250  -              .EQ ELAEFD=$AEFD
260  -              .EQ ELAF08=$AF08
270  -              .EQ ELE257=$E257
280  -              .EQ ELF5FA=$F5FA
290  -;
300  -              .BA $033C
310  -;
320  -              JSR ELAEFD
330  -              JSR ELE257
340  -              LDA #$08          ; "."
350  -              STA ELB8
360  -              JSR ELAEFD
370  -              BEQ L03C8
380  -              BCC L0398
390  -              CMP #$AB          ; "+"
400  -              BNE L03C8
410  -              JSR EL73
420  -              JSR L03C2
430  -              LDA EL2B
440  -              LDX EL2C
450  -              STA ELC1
460  -              STX ELC2
470  -L035F         LDY #$05          ; "."
480  -L0361         LDA (EL5F),Y
490  -              BEQ L036D
500  -              INC EL5F
510  -              BNE L0361
520  -              INC EL60
530  -              BNE L0361
540  -L036D         LDX EL60
550  -              LDA EL5F
560  -              ADC #$07          ; "."
570  -              BCC L0376
580  -              INX
590  -L0376         STA ELAE
600  -              STX ELAF
610  -              LDY #$06          ; "."
620  -              LDX #$00          ; "."
630  -              LDA (EL5F),Y
640  -              PHA
650  -              TXA
660  -              STA (EL5F),Y
670  -              INY
680  -              LDA (EL5F),Y
690  -              PHA
700  -              TXA
710  -              STA (EL5F),Y
720  -              JSR ELF5FA
730  -              LDY #$07          ; "."
740  -              PLA
750  -              STA (EL5F),Y
760  -              DEY
770  -              PLA
780  -              STA (EL5F),Y
790  -              RTS
800  -;
810  -L0398         JSR L03C2
820  -              LDA EL5F
830  -              LDX EL60
840  -              STA ELC1
850  -              STX ELC2
860  -              JSR EL73
870  -              BEQ L035F
880  -              CMP #$AB          ; "+"
890  -              BNE L03C8
900  -              JSR EL73
910  -              BEQ L03B7
920  -              JSR L03C2
930  -              JMP L035F
940  -;
950  -L03B7         LDA EL2D
960  -              LDX EL2E
970  -              STA ELAE
980  -              STX ELAF
990  -              JMP ELF5FA
1000 -;
1010 -L03C2         JSR ELA96B
1020 -              JMP ELA613
1030 -;
1040 -L03C8         JMP ELAF08

READY.
```

Hotwheels

ERTEM FIRAT

İşte size, BASIC ve makine dilinin ideal karışımından oluşan güzel, hızlı, hareketli ve heyecanlı bir oyun.

Amacınız, sokakların kıvrımlarına ve önünüze çıkan engellere aldırmadan, motosikletinizin üzerinde daha uzun kalabilmek, daha hızlı gidebilmek.

Programı yazın ve çalıştırın. Programın esas kısmını oluşturan makine dili bölümü hafızaya yerleşecek ve sonrasında da oyun başlayacaktır. Oyunu oynamak için, PORT 1'e takılı joystick kullanmanız gerekiyor. Ancak klavyenin sol üst köşesindeki dört tuşu da (1, 2, CTRL ve ok) kullanabilirsiniz. Kolun ileri itilmesi hızını artırırken, geri çekilmesi yavaşlatacaktır. Sokağın kıvrımlarına ve engellere karşı da ancak sağ-sol yaparak korunabilirsiniz. Hızınızı artırdıkça ekranın üst kenarına yaklaşacağınız için, önünüze çıkacak engelleri ve yolun değişimini gördükten sonra tepki göstermek için kullanabileceğiniz süre azalacak. İyi-si mi, dikkatli olun!

Eğer birinci aşamayı geçerseniz, ikinci aşamada sizi daha zor bir yol ve daha yüksek bir alt hız limiti bekliyor. Onu da geçersiniz, sırada üçüncü ve son aşama var. Puanlama, zorluk derecesine ve hızınıza göre ayarlanıyor. Amacınız, rekorlarınıza rekor katmak olmalı...

İsterseniz programa bir göz atalım. 1000-1040 arasında sprite okunuyor. 1050-1500 arasında da makine dili program 49152-50052 adresleri arasına yerleşiyor. 1590-1600'de de en sonda çalınacak melodinin notaları değişkenlere yükleniyor.

1610-1860 arası programın size merhaba deyişi ve oyuna ilk başlama.

1870-2080 arasında ekran, oyunun geçilen kademelerine göre yeniden hazırlanıyor. Ve 2090'da makine dili program çağırılarak oyun başlatılıyor.

2100-2130 arasında, makine dili programdan BASİC'e dönüş nedeninin çarpışma mı olduğu, yoksa bir sonraki kademeye mi geçileceği saptanıyor. 2140-2220 arası, ses yazmaçlarının tanımlanması kısmı. 2230'dan sonra da oyun sonu bildiriliyor, müzik çalıyor ve eğer varsa, yeni rekor gösteriliyor.

Şimdi makine dili programın depolama amacıyla kullandığı bazı adresleri görelim. Bunların bir kısmına istediğiniz sayıları koyarak programı kendinize göre değiştirebilirsiniz. Tıpkı BASIC programının 1820-1900 arası satırlarında yaptığı gibi...

832 : Cadde kenarının rengi.

870,871: Bir sonraki kademeye geçmek için gerekli puan.

872 : Engellerin ortaya çıkma sıklığı.

880,881: O anki puan durumu.

Tüm programda kullanılan değişkenler de şu şekilde:

A1(),A2(): Kapanış müziğinin notaları.

V : VIC'in temel adresi.

SI : SID'in temel adresi.

LE : O anki kademe

MR : Geriye kalan motosiklet sayısı.

HI : En son kırılmış rekor.

AS : Genel kullanımlı karakter değişkeni.

I,J,K. : FOR...NEXT döngü sayaçları.

İşte bu kadar... Yüksek rekorlar yapmanızı dileriz... □

PROGRAM ADI : HOTWHEELS

```
01000 DATA 0,16,0,0,16,0,0,16,0,0,32,0,1,169,0,4                <232>
01010 DATA 168,64,16,32,16,128,32,8,0,48,0,0,48,0,0,48          <038>
```

```
01020 DATA 0,0,252,0,0,252,0,0,252,0,0,48,0,0,48,0                    <142>
01030 DATA 0,16,0,0,16,0,0,00,0,0,00,0,0,0,0,0                        <109>
01040 RESTORE:FOR I=896 TO 959:READ A:POKE I,A:NEXT                   <005>
01050 DATA 076,181,193,076,012,192,076,099,192,076,214,194,162,00
      8,181,001,157                                                   <197>
01060 DATA 072,003,189,124,195,149,001,202,208,243,160,039,177,00
      6,145,008,177                                                   <085>
01070 DATA 002,145,004,136,016,245,162,007,181,001,056,233,040,14
      9,001,176,004                                                   <117>
01080 DATA 232,214,001,202,202,202,016,233,165,009,201,005,208,21
      8,165,008,201                                                   <233>
01090 DATA 024,208,212,162,008,189,072,003,149,001,202,208,248,09
      6,162,039,173                                                   <253>
01100 DATA 064,003,157,024,217,169,160,157,024,005,202,016,242,09
      6,032,082,192                                                   <015>
01110 DATA 160,012,169,012,024,109,065,003,170,169,032,157,024,00
      5,202,136,208                                                   <062>
01120 DATA 249,096,169,000,141,003,220,173,001,220,141,066,003,16
      2,004,169,000                                                   <065>
01130 DATA 157,067,003,078,066,003,062,067,003,202,016,244,169,00
      0,141,003,220                                                   <120>
01140 DATA 173,069,003,208,013,162,013,206,000,208,208,003,206,01
      6,208,202,208                                                   <128>
01150 DATA 245,173,068,003,208,013,162,013,238,000,208,208,003,23
      8,016,208,202                                                   <057>
01160 DATA 208,245,173,071,003,208,035,173,001,208,201,130,144,02
      8,206,092,003                                                   <090>
01170 DATA 162,011,206,001,208,202,208,250,238,082,003,024,173,08
      3,003,105,020                                                   <235>
01180 DATA 141,083,003,144,003,238,084,003,173,070,003,208,035,17
      3,001,208,201                                                   <142>
01190 DATA 221,176,028,238,092,003,162,011,238,001,208,202,208,25
      0,206,082,003                                                   <243>
01200 DATA 056,173,083,003,233,020,141,083,003,176,003,206,084,00
      3,173,082,003                                                   <253>
01210 DATA 141,001,212,173,83,3,141,007,212,173,084,003,141,008,2
      12,096,238                                                      <071>
01220 DATA 065,003,032,082,192,169,012,024,109,065,003,170,169,09
      5,157,024,005                                                   <182>
01230 DATA 202,160,011,169,032,157,024,005,202,136,208,249,169,22
      3,157,024,005                                                   <156>
01240 DATA 206,065,003,096,032,082,192,169,012,024,109,065,003,17
      0,169,233,157                                                   <102>
01250 DATA 024,005,202,160,011,169,032,157,024,005,202,136,208,24
      9,169,105,157                                                   <049>
01260 DATA 024,005,096,169,007,141,087,003,162,000,160,000,138,07
      2,152,072,032                                                   <185>
01270 DATA 012,192,173,031,208,240,004,104,076,100,194,032,121,19
      2,173,031,208                                                   <104>
01280 DATA 240,004,104,076,100,194,032,099,192,032,013,195,104,16
      8,104,170,189                                                   <217>
01290 DATA 102,194,153,052,005,232,189,102,194,153,052,217,232,20
      0,192,008,208                                                   <064>
01300 DATA 237,206,087,003,173,087,003,208,189,076,194,193,032,04
      0,195,173,065                                                   <172>
01310 DATA 003,201,008,208,003,076,104,193,234,234,234,172,027,21
      2,240,251,192                                                   <116>
01320 DATA 010,176,247,174,027,212,224,085,176,050,140,085,003,17
      3,065,003,056                                                   <246>
01330 DATA 237,085,003,144,237,152,072,032,214,194,032,012,192,17
      3,031,208,208                                                   <055>
01340 DATA 118,032,121,192,173,031,208,208,110,206,065,003,032,03
      2,193,032,013                                                   <075>
01350 DATA 195,104,168,136,208,221,076,181,193,224,170,176,035,15
      2,072,032,214                                                   <150>
01360 DATA 194,032,012,192,173,031,208,208,076,032,121,192,173,03
      1,208,208,068                                                   <193>
01370 DATA 032,099,192,032,013,195,104,168,136,208,224,076,181,19
      3,140,085,003                                                   <172>
01380 DATA 173,065,003,024,109,085,003,176,148,201,026,176,144,15
      2,072,032,214                                                   <085>
```

```
01390 DATA 194,032,012,192,173,031,208,208,025,032,121,192,173,03
      1,208,208,017                                              <254>
01400 DATA 238,065,003,032,071,193,032,013,195,104,168,136,208,22
      1,076,181,193                                              <080>
01410 DATA 104,096,160,005,160,002,160,002,160,002,160,002,160,00
      2,160,002,160                                              <139>
01420 DATA 005,160,005,160,002,160,002,160,002,160,002,160,002,16
      0,002,160,005                                              <038>
01430 DATA 160,005,160,002,091,006,160,002,160,002,091,006,160,00
      2,160,005,160                                              <020>
01440 DATA 005,160,002,160,002,160,002,160,002,160,002,160,002,16
      0,005,105,005                                              <062>
01450 DATA 032,001,032,001,032,001,032,001,032,001,032,001,095,00
      5,160,005,105                                              <180>
01460 DATA 005,032,001,032,001,032,001,032,001,095,005,160,005,16
      0,005,160,005                                              <031>
01470 DATA 160,005,160,005,160,000,160,005,160,005,160,005,162,00
      0,160,033,024                                              <035>
01480 DATA 032,010,223,173,112,003,024,105,016,144,003,238,113,00
      3,056,237,092                                              <233>
01490 DATA 003,141,112,003,176,003,206,113,003,173,113,003,174,11
      2,003,205,103                                              <112>
01500 DATA 003,208,009,236,102,003,144,004,104,104,104,096,032,20
      5,189,096,173                                              <100>
01510 DATA 092,003,141,093,003,169,001,141,094,003,206,095,003,20
      8,251,206,094                                              <226>
01520 DATA 003,208,246,206,093,003,208,236,096,173,027,212,201,00
      3,176,023,173                                              <208>
01530 DATA 065,003,105,004,170,160,003,169,127,157,024,005,169,00
      2,157,024,217                                              <196>
01540 DATA 232,136,208,242,096,205,104,003,176,043,173,027,212,04
      1,015,201,001                                              <145>
01550 DATA 240,247,168,173,027,212,201,003,144,249,201,012,176,24
      5,109,065,003                                              <163>
01560 DATA 170,152,157,024,217,173,027,212,201,128,176,008,169,09
      0,157,024,005                                              <120>
01570 DATA 076,124,195,169,081,157,024,005,096,152,219,192,219,15
      2,007,192,007                                              <083>
01580 FOR I=49152 TO 50052:READ A:POKE I,A:NEXT                  <013>
01590 DATA 103,17,237,21,20,26,69,29,3,31,69,29,20,26,237,21,103,
      17                                                         <236>
01600 FOR I=1 TO 9:READ A1(I),A2(I):NEXT                         <156>
01610 V=53248:POKE V,160:POKE V+1,215                            <244>
01620 POKE V+23,1:POKE V+29,1                                    <124>
01630 POKE V+28,1:POKE V+37,12:POKE V+38,0:POKE V+39,7           <253>
01640 POKE 2040,14:POKE 53281,2                                  <253>
01650 PRINT "<CLR><CTRL-8>";                                     <038>
01660 PRINT "<11SPACE><RVSON>£<RVSOFF><3SPACE><RVSON>£<RVSOFF> <R
      VSON>£<4SPACE><RVSOFF> <RVSON>£<3SPACE><RVSOFF>£"          <112>
01670 PRINT "<11SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <R
      VSON> <RVSOFF><3SPACE><RVSON> <RVSOFF><3SPACE><RVSON> "    <193>
01680 PRINT "<11SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <R
      VSON> <RVSOFF><3SPACE><RVSON> <RVSOFF><3SPACE><RVSON> "    <203>
01690 PRINT "<11SPACE><RVSON><5SPACE><RVSOFF> <RVSON> <RVSOFF><3S
      PACE><RVSON> <RVSOFF><3SPACE><RVSON> "                     <237>
01700 PRINT "<11SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <R
      VSON> <RVSOFF><3SPACE><RVSON> <RVSOFF><3SPACE><RVSON> "    <223>
01710 PRINT "<11SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <R
      VSON> <RVSOFF><3SPACE><RVSON> <RVSOFF><3SPACE><RVSON> "    <233>
01720 PRINT "<11SPACE>£<3SPACE>£ <RVSON><4SPACE><RVSOFF>£<3SPACE>
      £":PRINT :PRINT                                            <114>
01730 PRINT "<2SPACE><RVSON>£<RVSOFF><3SPACE><RVSON>£<RVSOFF> <RV
      SON>£<RVSOFF><3SPACE><RVSON>£<RVSOFF> <RVSON>£<3SPACE><RVSO
      FF>£ <RVSON>£<3SPACE><RVSOFF>£ <RVSON>£<RVSOFF><5SPACE><RVS
      ON>£<3SPACE><RVSOFF>£"                                     <057>
01740 PRINT "<2SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RV
      SON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RVSON> <RVSOFF><5SPA
      CE><RVSON> <RVSOFF><5SPACE><RVSON> <RVSOFF><5SPACE><RVSON>
      "                                                          <114>
01750 PRINT "<2SPACE><RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RV
      SON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RVSON> <RVSOFF><5SPA
```

```
      CE><RVSON> <RVSOFF><5SPACE><RVSON> <RVSOFF><5SPACE><RVSON>
      "                                                          <124>
01760 PRINT "<2SPACE><RVSON> <RVSOFF> <RVSON>£<RVSOFF> <RVSON> <R
      VSOFF> <RVSON><5SPACE><RVSOFF> <RVSON><3SPACE><RVSOFF>£<2SP
      ACE><RVSON><3SPACE><RVSOFF>£<2SPACE><RVSON> <RVSOFF><5SPACE
      ><RVSON><5SPACE>"                                          <001>
01770 PRINT "<2SPACE><RVSON> <RVSOFF> <RVSON> <RVSOFF> <RVSON> <R
      VSOFF> <RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RVSON> <RV
      SOFF><5SPACE><RVSON> <RVSOFF><5SPACE><RVSON> <RVSOFF><3SPAC
      E><RVSON> "                                                <120>
01780 PRINT "<2SPACE><RVSON> <RVSOFF> <RVSON> <RVSOFF> <RVSON> <R
      VSOFF> <RVSON> <RVSOFF><3SPACE><RVSON> <RVSOFF> <RVSON> <RV
      SOFF><5SPACE><RVSON> <RVSOFF><5SPACE><RVSON> <RVSOFF><3SPAC
      E><RVSON> "                                                <130>
01790 PRINT "<2SPACE><RVSON><4SPACE><RVSOFF>£ £<3SPACE>£ <RVSON><
      4SPACE><RVSOFF>£ <RVSON><4SPACE><RVSOFF>£ <RVSON><4SPACE><R
      VSOFF>£ <RVSON>£<3SPACE><RVSOFF>£"                         <220>
01800 PRINT :PRINT "<30SPACE>":PRINT                             <104>
01810 PRINT "<9SPACE>JOYSTICK ATESINE BASIN                       <011>
01820 POKE V+21,1:POKE 832,5:POKE 880,0:POKE 881,0:LE=1:MR=3:POKE
       198,0                                                     <193>
01830 POKE 53280,7                                               <116>
01840 POKE 53280,2                                               <116>
01850 GET A$:IF A$="" THEN 1830                                  <124>
01860 POKE V+23,0:POKE V+29,0                                    <108>
01870 IF MR=0 THEN 2230                                          <049>
01880 POKE 860,(4-LE)+12:POKE V+21,0                             <194>
01890 IF LE<3 THEN L1=LE*2000:POKE 870,L1 AND 255:POKE 871,L1/256 <184>
01900 POKE 872,LE*30:POKE V,180:POKE V+1,230:POKE 833,14         <110>
01910 PRINT "<CLR><CTRL-7>";:POKE 53281,1:POKE 53280,7:SI=54272:P
      OKE V+31,0                                                 <094>
01920 PRINT "<RVSON> REKOR=<20SPACE>PUAN=<8SPACE>";              <021>
01930 PRINT "<RVSON><10SPACE><RVSOFF><3SPACE><RVSON> <RVSOFF><2SP
      ACE><RVSON><24SPACE><RVSOFF>";                             <231>
01940 PRINT "<RVSON><6SPACE><RVSOFF><11SPACE><RVSON><6SPACE><RVSO
      FF><2SPACE><RVSON><5SPACE><RVSOFF><7SPACE><RVSON><3SPACE><R
      VSOFF>";                                                   <245>
01950 PRINT "<RVSON><5SPACE><RVSOFF><13SPACE><RVSON><4SPACE><RVSO
      FF><4SPACE><RVSON><3SPACE><RVSOFF><10SPACE>";              <251>
01960 PRINT "<RVSON><5SPACE><RVSOFF><11SPACE><RVSON><9SPACE><RVSO
      FF> <RVSON><13SPACE><RVSOFF>";                             <005>
01970 PRINT "<RVSON><6SPACE><RVSOFF><5SPACE><RVSON> <RVSOFF><3SPA
      CE><RVSON><24SPACE><RVSOFF>";                              <015>
01980 PRINT "<RVSON><40SPACE><RVSOFF>";                          <217>
01990 PRINT "<HOME><RVSON>"TAB(7)MID$(STR$(HI),2);               <088>
02000 PRINT TAB(16);:FOR I=1 TO MR:PRINT "Q ";:NEXT              <022>
02010 PRINT TAB(33)MID$(STR$(PEEK(880)+256*PEEK(881)),2)         <096>
02020 POKE SI+15,100:POKE SI+18,129                              <137>
02030 GOSUB 2140                                                 <104>
02040 SYS 49158                                                  <067>
02050 FOR I=1 TO 18:SYS 49155:NEXT                               <026>
02060 PRINT "<CTRL-3><HOME><8DOWN><17RIGHT>ZORLUK"LE             <155>
02070 FOR I=1 TO 18:SYS 49155:FOR J=1 TO 100:NEXT :NEXT          <156>
02080 POKE V+21,1                                                <209>
02090 PRINT "<RVSON>";:SYS 49152                                 <013>
02100 IF PEEK(880)+256*PEEK(881)<LE*2000 OR LE=3 THEN GOSUB 2200:
      MR=MR-1:GOTO 1870                                          <093>
02110 IF LE<3 THEN LE=LE+1:IF LE<3 THEN 1870                     <086>
02120 POKE 870,255:POKE 871,255:GOTO 1870                        <250>
02130 END                                                        <100>
02140 POKE SI,P:POKE SI+1,3:POKE 850,3:POKE SI+4,0               <187>
02150 POKE SI+5,255:POKE SI+6,255                                <215>
02160 POKE SI+7,80:POKE 851,80:POKE SI+8,0:POKE 852,0:POKE SI+11,
      0                                                          <091>
02170 POKE SI+12,255:POKE SI+13,255                              <057>
02180 POKE SI+4,33:POKE SI+11,129                                <230>
02190 POKE SI+24,138:RETURN                                      <180>
02200 POKE SI+4,0:POKE SI+11,0:POKE V+21,0:POKE SI+1,50:POKE SI+5
      ,0                                                         <241>
02210 POKE SI+6,250:POKE SI+4,129:POKE SI+4,28                   <047>
02220 FOR I=1 TO 200:POKE V+21,0:POKE V+21,1:NEXT :RETURN        <189>
```

```
02230 POKE 53280,13:POKE SI++11,0:POKE V+21,0:PRINT "<CLR>"          <119>
02240 FOR I=1 TO 9:POKE SI,A1(I):POKE SI+1,A2(I):POKE SI+4,17       <013>
02250 FOR J=1 TO 200:NEXT :POKE SI+4,16:NEXT                        <243>
02260 IF PEEK(880)+256*PEEK(881)<HI THEN 2265                       <183>
02261 GOTO 2270                                                     <177>
02265 PRINT "<HOME><DOWN><RIGHT>PUAN="PEEK(880)+256*PEEK(881):GOT
      O 1000                                                        <007>
02270 HI=PEEK(880)+256*PEEK(881)                                    <042>
02280 A$="........................................."                 <110>
02290 A$=A$+"YENI REKOR"+A$:POKE 53281,1:PRINT "<12DOWN><CTRL-4>"
      TAB(18)HI                                                     <155>
02300 FOR K=1 TO 2:FOR I=1 TO 59                                    <134>
02310 PRINT "<HOME><9DOWN><CTRL-4>"MID$(A$,I,40)                    <170>
02320 PRINT "<HOME><10DOWN><C=3>"MID$(A$,60-I,40)                   <252>
02330 FOR J=1 TO 100:NEXT :NEXT :NEXT :GOTO 1000                    <127>
```

Diskfaster

TUNA ERTEMALP

VC1541 disket sürücüsünün yavaş olduğunu iddia edenler! İşte onu hızlandırmanızı sağlayacak bir program. LOAD işlemleri 5-6 kat, SAVE işlemleri de 3-5 kat daha hızlanabilir...

Önce programı MDE ile yazın ve diskete kaydedin. Çalıştırmak istediğiniz zaman, bilgisayara yükleyip RUN ile çalıştırın. Bilgisayar, açılış ekranına benzer bir ekranı gösterecek ve bu arada da hızlı SAVE ve LOAD rutinleri hafızaya yerleşmiş olacaktır. Bu açılış ekranı sırasında, kaç byte kullanım alanı kaldığı gösterilecektir. Bazen bu sayı yanlış olabilir, dikkate almayın. Şimdi her iki komutla ilgili bazı açıklamalar yapalım.

Hızlı LOAD rutinleri, $A000-$BFFF arasındaki ROM'un altına yerleşmektedir. Eğer normal LOAD hızına dönmek isterseniz, POKE 1,55 vermeniz yeterli olacaktır. Ayrıca demin belirttiğimiz iki adres arasını çeşitli amaçlar için kullanmak zorundaysanız, kullanmadan önce deminki POKE komutunu mutlaka kullanmalısınız. Hızlı LOAD rutinlerini POKE 1,55 ile kapattıktan sonra yeniden aktif hale getirmek isterseniz, eğer bu arada $A000-$BFFF arasına dokunmadıysanız, POKE 1,53 komutu ile olacaktır. RUN-STOP/RESTORE tuşlarına basmanız, hızlı LOAD rutinlerini etkilemeyecektir.

Eğer yüklenen bir program, daha yüklenirken $A000 sonrasına geçiyorsa, yeni LOAD rutinleri otomatik olarak bunun farkına varacak ve gerektiği zaman aradan çekilip, ROM'da duran ve bu nedenle yüklenen programın uzunluğundan etkilenmeyen normal LOAD rutinlerini çağıracaktır.

Buna bir istisna, $E000 sonrasında yeralması gereken programlar. Bunlar belli bir uzunluğu geçtiklerinde, hızlı LOAD rutininin çalışma alanını zedelemektedirler. Bu, bilgisayarınızın kilitlenmesine neden olabilir. Bundan başka, kimi programlar hızlı LOAD rutini ile yüklenirken sorun çıkarmamakta, ancak RUN komutu verildiğinde çalışmamaktadırlar. Örneğin disket/kaset üzerinde aldığınız SIMONS' BASIC'in kimi türleri böyledir. Bu durumda LOAD işlemi bitince POKE 1,55 ile eski rutinleri aktif hale getirmek, sonra RUN yapmak gerekmektedir.

Bir başka nokta da, hızlı LOAD rutinleri aktif iken, kaset ve RS232 çıkışını kullanan işlemlerin yapılamayacağı. Bunlardan önce de, POKE 1,55 ile yeni rutinler aradan çıkartılmalı.

Yukarıda, RUN-STOP/RESTORE'nin yeni rutinleri bozmadığını anlatmıştık. Aynı şekilde SYS 64738 ile yapılacak bir reset de, programı doldurduğunuz zamanki açılış ekranına benzer bir ekranın görünmesine ve hızlı LOAD rutinlerinin çalışırlıklarını korumasına yolaçacaktır. Eğer bir RESET SWITCH tuşu kullanıyorsanız, ona bastıktan sonra POKE 1,53 ile yeni rutinleri, eğer bozulmamışlarsa, yeniden aktif hale getirebilirsiniz.

LOAD işlemi sırasında, ekrandaki görüntü geçici bir süre için yok olacaktır. Yükleme bitince yeniden görünecektir. Bu arada sizin dikkat etmeniz gereken şey, LOAD işlemi sırasında printer'ın açık olmaması. Eğer açıksa, yeni rutinler sizi uyaracaktır. Bu durumda printer'ı kapatıp RUN/STOP'a basın. Yükleme işlemi devam edecektir.

Hızlı SAVE rutinlerine gelince... DISKFASTER'i RUN ile çalıştırdığınızda karşınıza çıkan açılış ekranının bir köşesinde de SYS 365 yazdığını göreceksiniz. Bu, hızlı LOAD rutinlerini aktif hale getiren komuttur. Eğer reset, SYS 64738 veya RUN-STOP/RESTORE gibi nedenlerle bu rutinlerin çalışırlığı ortadan kalkmışsa, SYS 365 ile yeniden başlatabilirsiniz.

Yeni rutinlerin ilginç bir özelliği var. Normal rutinler, bir programı diskete SAVE ettikten sonra, bir de kontrol işlemi yaparlar. Bu, zaman

kaybına yolaçar. Eğer çok kaliteli disket kullanıyorsanız, disket sürücünüz de hata yapmıyorsa veya programın diskete hatalı kaydedilmiş olması sizi üzmeyecekse, ki hiç sanmıyoruz, yeni rutinler kontrolu iptal etme olanağını veriyor. Normal SAVE komutunu vererek kayıt işlemi yaptığınızda, kontrol yapılacaktır. Ancak programın adının başına * (çarpı) işareti koyarsanız, kontrol iptal edilecektir. Örneğin:

```
SAVE "deneme",8
SAVE "*deneme",8
```

Bir başka yenilik de, programın SAVE edilmesi biter bitmez, yani bilgisayarın disket sürücüsüne gönderecek bilgisi kalmadığında, bilgisayarın kendi işine geri dönmesidir. Bunu, SAVE işlemi görünürde bittiği halde disket sürücüsünün yanan kırmızı ışığından anlayabilirsiniz. Bu sırada disket sürücüsü dosyanın kapatılması ile ilgili işlemler yapmaktadır. Bu, bilgisayarı ilgilendirmediği için, bilgisayar kendi işine devam etmektedir. Eskiden bilgisayar, disket sürücüsünün bu işlemleri de bitirmesini bekliyordu. Ancak, disketi disket sürücüsünün içinden çıkarmak için kırmızı ışığın sönmesini bekleyeceksiniz.

Hızlı SAVE işlemi yapılırken, kayıdı durdurmak isterseniz RESTORE tuşuna basmalısınız. Ancak bu durumda disket üzerindeki dosya kapatılmayacak ve directory'de bir yıldız işareti ile görünecektir. Ayrıca, kırmızı ışık da yanıp sönecektir. Işık önemli değil ama, bu disketi sonradan OPEN 1,8,15,"V":CLOSE 1 komutunu göndererek temizlemeniz gerekir (diğer dosyalara bir şey olmayacaktır.)

Eğer disket sürücüsü bir işlem sonucunda 61 veya 71 numaralı hataları gönderiyorsa, disket sürücüsünü kapatıp açmaktan başka çare yok demektir.

DISKFASTER rutinleri, SX-64'ün donanımıyla kimi zaman sorun çıkarmaktadır. Eğer çalışmazsa sakın şaşırmayın. Ancak VC1541 ile çalışma kusursuz olmaktadır. C128 ve VC1571 kullananlara gelince. C128'i C64 modunda kullandığınız sürece DISKFASTER rutinlerinin bir sorun çıkarmaması lazım. Bugüne kadar böyle bir soruna rastlanmadı. Ancak C128'i C128 veya CP/M moduna geçirdiğinizde DISKFASTER çalışmayacaktır.

Tüm çalışmalarınızda kolaylık sağlayacağını ümit ederiz...

```
PROGRAM  : DISKFASTER          0801 137F
-----------------------------------------

0801 : 0B 08 0A 00 9E 32 30 36   3C
0809 : 31 00 00 00 A9 2B A0 08   C1
0811 : 85 5F 84 60 A9 7F A0 13   B2
0819 : 85 5A 84 5B A9 54 A0 9B   4F
0821 : 85 58 84 59 20 BF A3 4C   46
0829 : 00 90 A9 63 A0 90 85 5F   AB
0831 : 84 60 A9 A2 A0 95 85 5A   25
0839 : 84 5B A9 40 A0 0D 85 58   16
0841 : 84 59 20 BF A3 20 0E 08   F5
0849 : A9 A2 A0 95 85 5F 84 60   44
0851 : A9 54 A0 9B 85 5A 84 5B   B4
0859 : A9 B3 A0 0D 85 58 84 59   85
0861 : 20 BF A3 A9 53 8D 77 02   02
0869 : A9 59 8D 78 02 A9 53 8D   07
0871 : 79 02 A9 33 8D 7A 02 A9   C4
0879 : 36 8D 7B 02 A9 35 8D 7C   08
0881 : 02 A9 0D 8D 7D 02 A9 07   E9
0889 : 85 C6 20 A6 0C 0C 08 C1   13
0891 : 07 9E 20 32 30 36 32 00   B3
0899 : 00 00 78 A5 01 48 A9 34   CD
08A1 : 85 01 A0 28 B9 51 08 99   4D
08A9 : 4F 01 88 D0 F7 A2 06 84   6B
08B1 : AC A9 D0 85 AD A9 A0 85   CC
08B9 : 14 A9 08 85 15 B1 14 91   A7
08C1 : AC C8 D0 F9 E6 15 E6 AD   53
08C9 : CA D0 F2 20 6D 01 68 85   48
08D1 : 01 58 60 1E 90 A0 07 20   45
08D9 : D2 FF E8 D0 F7 60 A5 BA   8D
08E1 : C9 04 B0 03 4C ED F5 78   36
08E9 : A6 01 A9 34 85 01 8E 9F   DB
08F1 : D4 20 4F D4 AE 9F D4 86   8C
08F9 : 01 58 60 A9 50 8D 32 03   B4
0901 : A9 01 8D 33 03 60 0D 0D   76
0909 : 09 0E 20 20 20 20 20 20   E9
0911 : 20 20 20 20 20 20 20 20   11
0919 : 20 20 20 20 20 20 20 20   19
0921 : 20 20 20 20 20 20 20 20   21
0929 : 20 20 20 0D BA 8E A0 D4   4F
0931 : AD 19 03 8D A1 D4 AD 18   85
0939 : 03 8D A2 D4 A9 9C 8D 18   2C
0941 : 03 A9 C1 8D 19 03 AD 9F   DA
0949 : D4 85 01 AD 00 DD 8D 1F   39
0951 : 01 58 A9 61 85 B9 A2 2C   1E
0959 : A0 00 B1 BB C9 2A D0 0A   22
0961 : A2 4C C6 B7 E6 BB D0 02   65
0969 : E6 BC 8E A5 C3 A9 C1 48   27
0971 : A9 75 48 A5 B7 D0 03 4C   42
0979 : 10 F7 20 D5 F3 20 8F F6   B4
0981 : A5 BA 20 0C ED A5 B9 20   40
0989 : B9 ED A9 00 85 90 20 DD   BC
0991 : ED 20 DD ED 20 FE ED A5   C0
0999 : 90 F0 02 18 60 A9 1A A2   26
09A1 : C2 85 AC 86 AD A9 46 A2   A8
09A9 : 01 85 14 86 15 A2 05 A5   08
09B1 : BA 20 0C ED A9 6F 20 B9   46
09B9 : ED A0 FD B9 15 C1 20 DD   48
09C1 : ED C8 D0 F7 A5 14 20 DD   7D
09C9 : ED A5 15 20 DD ED A9 1E   02
09D1 : 20 DD ED A0 00 B1 AC 20   F0
09D9 : DD ED C8 C0 1E 90 F6 20   79
09E1 : FE ED A5 AC 18 69 1E 85   25
09E9 : AC 90 02 E6 AD A5 14 18   C3
09F1 : 69 1E 85 14 90 02 E6 15   2C
09F9 : CA D0 B4 A5 BA 20 0C ED   C6
0A01 : A9 6F 20 B9 ED A0 FB B9   E8
0A09 : 1A C1 20 DD ED C8 D0 F7   20
0A11 : 20 FE ED 78 A2 00 8E 00   9F
0A19 : DD AD 11 D0 29 EF 8D 11   96
0A21 : D0 BD AE C2 9D 00 C5 E8   96
0A29 : D0 F7 AD 00 DD 30 FB 20   F0
0A31 : FE C1 A2 00 BD AE C3 9D   54
0A39 : 00 C5 E8 D0 F7 20 FE C1   70
0A41 : 20 8E FB A2 01 A5 AC 9D   27
0A49 : 01 C5 E8 A5 AD 9D 01 C5   73
0A51 : E8 20 D1 FC B0 38 A5 AD   1C
0A59 : C9 C0 90 1D C9 C6 B0 19   12
0A61 : 69 16 85 AD A9 34 85 01   41
0A69 : B1 AC 48 AD 9F D4 85 01   F1
0A71 : A5 AD E9 15 85 AD 68 B0   D3
0A79 : 02 B1 AC 9D 01 C5 E8 20   55
0A81 : DB FC E0 FF 90 CB 20 D1   3E
0A89 : FC B0 03 A9 FF 2C A9 00   DB
0A91 : 48 8D 00 C5 8E 01 C5 20   A1
0A99 : FE C1 A2 01 68 D0 B2 18   49
0AA1 : A9 00 48 AD 11 D0 09 10   EE
0AA9 : 8D 11 D0 AD 1F 01 8D 00   D9
0AB1 : DD 78 A9 34 85 01 AD A2   18
0AB9 : D4 8D 18 03 AD A1 D4 8D   11
0AC1 : 19 03 68 AE A0 D4 9A 60   28
0AC9 : 38 AD 1F 01 09 10 8D 1F   45
0AD1 : 01 B0 CD B9 00 C5 85 95   44
0AD9 : A2 00 2C 00 DD 50 FB 2C   2F
0AE1 : 00 DD 10 05 E8 D0 F8 F0   4F
0AE9 : B6 A9 10 8D 00 DD AA 66   90
0AF1 : 95 6A 66 95 6A 4A 4A 8D   45
```

```
0AF9 : 00 DD 8A 66 35 6A 66 95   C9
0B01 : 6A 4A 4A 8D 00 DD 8A 66   BA
0B09 : 35 6A 66 95 6A 4A 4A 8D   5D
0B11 : 00 DD 8A 66 35 6A 66 35   E1
0B19 : 6A 4A 4A 8D 00 DD EA EA   5D
0B21 : EA A9 00 8D 00 DD C8 D0   45
0B29 : AA 60 A0 00 98 53 00 C5   0B
0B31 : C8 D0 FA 85 14 20 A7 C1   35
0B39 : 88 A5 14 4C AA C1 4D 2D   6B
0B41 : 57 4D 2D 45 B7 01 A0 00   39
0B49 : 84 11 A9 02 8D 00 18 A9   8D
0B51 : 04 2C 00 18 F0 FB A9 00   64
0B59 : 8D 00 18 A2 03 CA D0 FD   06
0B61 : A2 0A AD 00 18 4A 6A 4A   86
0B69 : 66 85 0A 0A 66 85 AD 00   9F
0B71 : 18 4A 6A 4A 66 85 0A 0A   61
0B79 : 66 85 AD 00 18 4A 6A 4A   1F
0B81 : 66 85 0A 0A 66 85 AD 00   B7
0B89 : 18 8E 00 18 4A 8A 4A 66   D9
0B91 : 85 0A 0A 66 85 A5 85 91   29
0B99 : 30 45 11 85 11 C8 D0 AA   51
0BA1 : 88 60 85 31 20 46 01 B1   7C
0BA9 : 30 48 20 4A 01 68 91 30   49
0BB1 : A5 11 F0 ED 4C 43 E8 78   4C
0BB9 : A9 0A 8D 00 18 A2 00 88   72
0BC1 : D0 FD CA D0 FA A9 0A 85   8D
0BC9 : 69 A9 00 85 30 A9 03 20   54
0BD1 : A2 01 A9 04 20 A2 01 4C   92
0BD9 : 3D 04 A5 00 A2 01 86 00   CE
0BE1 : 29 02 F0 10 A6 38 86 32   F7
0BE9 : 20 2E 04 20 AF 03 20 03   BF
0BF1 : 04 4C 69 F9 A2 08 20 35   0A
0BF9 : 03 A2 0A 20 35 03 A2 08   DA
0C01 : 20 F5 03 A2 0A 20 F5 03   B0
0C09 : A5 8C 30 E8 4C 69 F9 86   22
0C11 : 32 86 38 A5 8C 10 61 B5   9B
0C19 : 83 D0 5D BD 31 04 20 A2   0C
0C21 : 01 C8 A6 32 A9 80 95 83   72
0C29 : A5 80 95 00 A5 81 95 01   32
0C31 : B1 30 F0 1C 20 21 F1 A6   DA
0C39 : 82 F6 B5 D0 02 F6 BB A0   C6
0C41 : 00 A5 80 31 30 C8 A5 81   49
0C49 : 31 30 A5 80 C5 22 F0 02   A1
0C51 : 84 8C A6 32 A9 00 85 30   1C
0C59 : 85 33 85 2E 85 36 85 0C   D7
0C61 : 85 50 A9 BB 85 34 BD 31   43
0C69 : 04 85 2F 20 E9 F5 85 3A   D9
0C71 : BD 32 04 20 A3 F7 A6 32   45
0C79 : B5 83 F0 52 20 2E 04 AD   55
0C81 : 00 1C 29 10 D0 03 4C 81   35
0C89 : F5 20 10 F5 A2 09 50 FE   03
0C91 : B8 CA D0 FA A9 FF 8D 03   19
0C99 : 1C AD 0C 1C 29 1F 09 C0   44
0CA1 : 8D 0C 1C A9 FF A2 05 8D   B5
0CA9 : 01 1C B8 50 FE B8 CA D0   73
0CB1 : FA A0 BB B1 0C 50 FE B8   D1
0CB9 : 8D 01 1C C8 D0 F5 B1 30   CB
0CC1 : 50 FE B8 8D 01 1C C8 D0   26
0CC9 : F5 50 FE 4C 00 FE 60 86   B6
0CD1 : 32 2C 24 04 86 98 B5 83   AE
0CD9 : F0 F4 20 2E 04 20 0A F5   67
0CE1 : A0 BB B1 0C 50 FE B8 4D   C7
0CE9 : 01 1C D0 19 C8 D0 F3 B1   96
0CF1 : 30 50 FE B8 4D 01 1C D0   0F
0CF9 : 0C C8 C0 FD D0 F1 A6 32   F5
0D01 : A9 00 95 83 60 4C C5 F6   EE
0D09 : BD 31 04 35 31 BD 32 04   E2
0D11 : 85 0D 60 05 01 06 04 58   D7
0D19 : 20 19 F1 A9 84 D5 A7 F0   EF
0D21 : 05 95 A7 20 42 D0 A9 40   B1
0D29 : 8D F9 02 A9 01 85 83 20   F3
0D31 : 07 D1 90 03 4C F8 CF 20   B1
0D39 : 3E DE F6 B5 A9 00 85 88   22
0D41 : 85 8D A9 80 85 8C A5 30   5C
0D49 : 85 06 A9 E0 85 00 A5 00   47
0D51 : 30 FC F0 24 C9 01 F0 EA   FE
0D59 : A5 13 85 06 A5 13 85 07   74

0D61 : A2 00 A3 B0 20 7D D5 20   09
0D69 : 93 D5 A9 E2 20 7D D5 20   33
0D71 : 99 D5 A5 8C D0 CC F0 CE   C4
0D79 : 4C 23 D8 A0 C0 A2 D6 20   1E
0D81 : 6A D4 A0 D0 A2 C0 20 6A   10
0D89 : D4 20 00 C0 48 A0 D6 A2   80
0D91 : C0 20 6A D4 68 60 AD FA   CD
0D99 : FF 48 AD FB FF 48 A9 FE   88
0DA1 : 8D FA FF A9 D3 8D FB FF   7A
0DA9 : 84 AD 86 15 A0 00 84 AC   BE
0DB1 : 84 14 A2 06 B1 AC 31 14   98
0DB9 : C8 D0 F9 E6 AD E6 15 CA   40
0DC1 : D0 F2 68 8D FB FF 68 8D   53
0DC9 : FA FF 60 00 0B 08 C0 07   DD
0DD1 : 9E 33 32 33 38 00 00 00   7F
0DD9 : 20 2F F3 A9 E2 A2 F8 85   F5
0DE1 : A7 86 A8 A9 00 A2 03 85   57
0DE9 : A9 86 AA A9 08 20 0C ED   43
0DF1 : A9 6F 20 B9 ED A9 4D 20   33
0DF9 : DD ED A9 2D 20 DD ED A9   D9
0E01 : 57 20 DD ED A0 00 A5 A9   91
0E09 : 20 DD ED A5 AA 20 DD ED   47
0E11 : A9 1E 20 DD ED B1 A7 20   D8
0E19 : DD ED C8 C0 1E 90 F6 20   B9
0E21 : FE ED 18 A5 A7 69 1E 85   1A
0E29 : A7 90 03 E6 A8 18 A5 A9   EB
0E31 : A6 AA 69 1E 85 A9 30 02   36
0E39 : E6 AA E0 05 90 AD C9 00   EB
0E41 : 90 A9 A9 08 20 0C ED A9   7F
0E49 : 6F 20 B9 ED A9 4D 20 DD   36
0E51 : ED A9 2D 20 DD ED A9 45   E1
0E59 : 20 DD ED A9 3B 20 DD ED   25
0E61 : A9 04 20 DD ED A9 0B 8D   43
0E69 : 11 D0 20 FB ED EA EA 4C   E4
0E71 : 2E F0 A9 0B 8D 00 DD 2C   8C
0E79 : 00 DD 10 FB A9 03 8D 00   D4
0E81 : DD A2 05 CA EA D0 FC A2   B8
0E89 : 04 AD 00 DD 2A 2A 66 B0   0F
0E91 : 6A 66 B0 EA CA D0 F2 A5   02
0E99 : B0 49 FF 60 20 52 EF C9   E2
0EA1 : FF F0 F8 A0 00 A9 0B 8D   FF
0EA9 : 00 DD 2C 00 DD 10 FB A9   45
0EB1 : 03 8D 00 DD A2 07 CA D0   66
0EB9 : FD AD 00 DD 2A 2A 66 B0   38
0EC1 : 6A 66 B0 EA EA AD 00 DD   C0
0EC9 : 2A 2A 66 B0 6A 66 B0 EA   2A
0ED1 : EA AD 00 DD 2A 2A 66 B0   3D
0ED9 : 6A 66 B0 EA EA AD 00 DD   D8
0EE1 : 2A 2A 66 B0 6A 66 B0 A5   B8
0EE9 : B0 49 FF 99 A6 FB C8 D0   30
0EF1 : B4 60 78 A9 01 85 A7 A0   45
0EF9 : FF 20 7C EF C0 FF F0 48   76
0F01 : A2 02 A5 A7 F0 02 A2 04   B4
0F09 : AD A6 FB D0 07 EE A7 FB   A1
0F11 : AD A7 FB 2C A9 00 85 A8   18
0F19 : BD A6 FB 31 AE E6 AE D0   D3
0F21 : 02 E6 AF E8 E4 A8 D0 F0   58
0F29 : A2 00 86 A7 AD A6 FB D0   03
0F31 : C6 A9 35 85 01 A9 1B 8D   AF
0F39 : 11 D0 A9 40 85 90 18 60   23
0F41 : A9 35 85 01 A2 1B 8D 11   D2
0F49 : D0 EA A9 1D 38 60 A0 00   A5
0F51 : B9 42 F0 99 30 01 C8 C0   4A
0F59 : 1F D0 F5 20 D2 EF 4C 30   A0
0F61 : 01 EA 08 48 A9 00 A8 53   D2
0F69 : 00 A0 C8 D0 FA C9 30 F0   E7
0F71 : 04 A9 37 85 01 A6 AE A4   12
0F79 : AF 68 28 58 60 DD 10 AD   02
0F81 : F0 22 AD A1 02 4A B0 FA   4D
0F89 : AD 01 DD 29 FD 8D 01 DD   5F
0F91 : AD 01 DD 29 04 F0 F9 A9   5E
0F99 : 30 18 4C 3B EF A0 A1 02   A7
0FA1 : 23 12 F0 F3 18 60 AD 37   E8
0FA9 : 02 AC 3C 02 CC 3B 02 F0   F0
0FB1 : 0B 29 F7 8D 37 02 B1 FB   41
0FB9 : EE 3C 02 60 09 08 8D 37   B0
0FC1 : 02 A9 00 60 48 AD A1 02   B0
```

```
0FC9 : F0 11 AD A1 02 29 03 D0 F8
0FD1 : F3 A3 10 8D 0D DD A3 00 BE
0FD9 : 8D A1 02 68 60 A5 00 29 4A
0FE1 : 06 C9 02 F0 03 4C 3E FD 74
0FE9 : EA A9 05 85 09 A2 5A 86 B6
0FF1 : 4B A2 00 A9 52 85 24 20 E5
0FF9 : 56 F5 50 FE B8 AD 01 1C 72
1001 : C5 24 F0 03 C6 4B D0 EF 1F
1009 : A9 0A 4C 69 F3 50 FE B8 87
1011 : AD 01 1C 95 25 E8 E0 07 24
1019 : D0 F3 20 97 F4 A5 16 45 3D
1021 : 17 45 18 45 19 45 1A F0 90
1029 : 07 C6 09 D0 C0 4C 1E F4 C1
1031 : A5 18 C5 06 F0 03 4C 0B 83
1039 : F4 85 22 A9 06 85 31 4C 98
1041 : 3C 04 A5 12 A6 13 85 16 70
1049 : 86 17 A5 06 85 18 A5 07 43
1051 : 85 19 A9 00 45 16 45 17 16
1059 : 45 18 45 19 85 1A 20 34 31
1061 : F9 A2 5A 20 56 F5 A0 00 DE
1069 : 50 FE B8 AD 01 1C D9 24 BD
1071 : 00 F0 06 CA D0 ED 4C 51 14
1079 : F5 C8 C0 08 D0 EA 20 56 35
1081 : F5 50 FE B8 AD 01 1C 91 EB
1089 : 30 C8 D0 F5 A0 BA 50 FE 2F
1091 : B8 AD 01 1C 99 00 01 C8 13
1099 : D0 F4 20 E0 F8 A5 38 C5 31
10A1 : 47 F0 03 4C F6 F4 20 E9 16
10A9 : F5 C5 3A F0 03 4C 02 F5 B4
10B1 : A0 00 A9 55 20 52 04 B9 7E
10B9 : 00 06 85 77 2C 00 18 10 50
10C1 : FB A9 10 8D 00 18 2C 00 B8
10C9 : 18 30 FB A2 00 8A 66 77 29
10D1 : 2A 2A 66 77 2A 2A 8D 00 C3
10D9 : 18 8A 66 77 2A 2A 66 77 3B
10E1 : 2A 2A 8D 00 18 8A 66 77 E2
10E9 : 2A 2A 36 77 2A 2A 8D 00 DB
10F1 : 18 8A 66 77 2A 2A 66 77 53
10F9 : 2A 2A 8D 00 18 A2 02 CA D0
1101 : D0 FD A9 0F 8D 00 18 C8 E7
1109 : D0 AD EA EA EA EA EA EA 4F
1111 : EA AD 00 1C 09 08 8D 00 50
1119 : 1C AD 00 06 D0 03 4C 9E 60
1121 : FD C5 18 D0 F9 85 06 AD 60
1129 : 01 06 85 07 4C 65 03 85 77
1131 : 77 2C 00 18 10 F5 A9 10 69
1139 : 8D 00 18 2C 00 18 30 FB C8
1141 : A2 04 A9 00 66 77 2A 2A 6F
1149 : 66 77 2A 2A 8D 00 18 CA 0A
1151 : D0 F0 EA EA EA EA EA EA 39
1159 : A9 0F 8D 00 18 60 60 85 FE
1161 : 00 58 A5 00 30 FC 78 60 94
1169 : 78 EA EA EA EA EA EA A5 68
1171 : 19 8D 00 06 85 06 A5 19 62
1179 : 8D 01 06 85 07 A9 04 85 92
1181 : 78 A9 E2 20 82 04 C9 02 FE
1189 : 90 33 A0 00 84 78 A4 78 88
1191 : B9 DB FE F0 12 58 20 76 67
1199 : D6 78 A9 E2 20 82 04 C9 2C

11A1 : 02 90 1A E6 78 D0 E7 A9 50
11A9 : C0 20 82 04 A9 E2 20 82 D2
11B1 : 04 C9 02 90 08 A9 FF 20 3B
11B9 : 52 04 4C 22 EB AD 00 06 9D
11C1 : F0 F8 C5 18 F0 C4 AD 00 8E
11C9 : 06 85 06 AD 01 06 85 07 2E
11D1 : 4C A0 04 EA EA EA EA A0 BF
11D9 : 00 B9 19 F4 99 30 01 C8 48
11E1 : C0 1F D0 F5 4C 30 01 A9 C1
11E9 : 00 A8 59 00 A0 C8 D0 FA 1D
11F1 : C9 80 F0 07 A9 37 85 01 84
11F9 : 4C 01 F5 4C 30 F4 A0 00 FA
1201 : B1 BB C9 24 F0 F2 A9 01 D6
1209 : 85 A7 A9 00 85 90 A5 A7 8F
1211 : 20 0C ED A9 6F 20 B9 ED A3
1219 : A5 90 10 0B E6 A7 A5 A7 FD
1221 : C9 10 D0 E6 4C BB EE A5 AD
1229 : A7 C9 08 F0 EF A0 00 B9 4C
1231 : 7A F4 F0 06 20 D2 FF C8 4C
1239 : D0 F5 20 E1 FF D0 FB EE 9C
1241 : EA EA EA EA 4C BB EE EA EC
1249 : 0D 4C 55 54 45 45 4E 20 64
1251 : 53 41 44 45 43 45 20 46 6A
1259 : 4C 4F 50 50 59 20 41 43 8D
1261 : 49 4B 20 21 0D 00 00 00 4D
1269 : 00 00 00 00 00 00 00 00 6A
1271 : 00 78 A9 37 85 01 A0 00 E1
1279 : 84 03 A9 A0 85 04 B1 03 42
1281 : 91 03 E6 03 D0 F8 E6 04 26
1289 : D0 F4 A9 E5 8D D6 FD A9 D5
1291 : 4C 8D 72 FE 8D F9 F4 A9 F0
1299 : BC 8D 73 FE A9 FE 8D 74 8A
12A1 : FE A9 09 8D FA F4 A9 F4 50
12A9 : 8D FB F4 A9 12 85 03 A9 53
12B1 : 0A 85 04 A9 E2 85 05 A9 76
12B9 : F8 85 06 B1 03 91 05 C8 8E
12C1 : D0 F9 E6 04 E6 06 A5 06 93
12C9 : C9 FA D0 EF A9 10 A2 08 F7
12D1 : 85 03 86 04 A9 BB A2 EE DB
12D9 : 85 05 86 06 B1 03 91 05 C7
12E1 : C8 D0 F9 E6 04 E6 06 A5 47
12E9 : 06 C9 F0 D0 EF B9 43 0D 1E
12F1 : 99 5F E4 C8 C0 50 D0 F5 4A
12F9 : A0 00 B9 0C 0C 99 09 F4 25
1301 : C8 C0 94 D0 F5 A9 35 85 55
1309 : 01 4C F8 FC EA EA 00 20 54
1311 : 42 59 54 45 53 20 46 52 B2
1319 : 45 45 0D 00 00 00 00 00 44
1321 : 00 00 93 0D 20 20 20 2A 80
1329 : 2A 2A 2A 20 43 36 34 2D 06
1331 : 44 49 53 4B 46 41 53 54 BC
1339 : 45 52 2D 53 49 53 54 45 68
1341 : 4D 49 20 2A 2A 2A 2A 0D 37
1349 : 0D 20 20 20 43 4F 4D 4D F1
1351 : 4F 44 4F 52 45 20 36 34 77
1359 : 20 20 00 81 48 20 00 00 3F
1361 : 00 00 00 00 00 00 00 00 62
1369 : 00 00 00 00 31 36 35 2C 58
1371 : 31 37 36 2C 2B 37 33 2C E3
1379 : 32 35 35 2C 30 39 2C 32 F8
```

Data Yaratıcı

ERTEM FIRAT

Bilgisayarlarca "utility" olarak adlandırılan, yardımcı programlar silsilesinden bir tanesine bu sayımızda yer veriyoruz. Maksat kolaylık olsun...

Diyelim ki yabancı dergilerden birinde veya bizim dergimizde, makine dilinde yazılmış bir program gördünüz ve o dergide verilen bir editör sayesinde yazdınız. Veya bambaşka bir yerden elinize RUN ile değil de, SYS komutuyla çalışan bir program geçti. Şimdi de bu programı başka bir yere SAVE etmek veya bir arkadaşınıza vermek istiyorsunuz. Eğer bir makine dili monitörünüz yoksa işiniz zor. Normal SAVE komutu ile belli bir hafıza bölümünü diskete veya kasete gönderemezsiniz. O zaman, ne yapmalı?..

Bunun birçok yolu var. Bunlardan biri de makine dilindeki programı DATA satırlarından oluşan bir BASIC programı haline çevirmek. Ama, oturup da bilmem kaç kilobyte'lık programı tane tane DATA'lara dönüştürmek, her babayiğitin harcı değil. Hem biz o bilgisayarı boşuna mı aldık. Bırakalım o yapsın bu işi.

Bizim programımızı kullanmak için, önce DATA satırlarına dönüştürülecek programı, 8,1 ile disketten veya, 1,1 ile kasetten yükleyin. Başlangıç adresini bilmeniz gerekiyor. Bitiş adresini de:

PRINT PEEK(174)+PEEK(175)*256

ile öğrenebilirsiniz.

Sonra da programımızı yükleyin ve run 500 ile çalıştırın. Aman dikkat!.. Sakın tek başına RUN demeyin, 500 yazmayı unutmayın!! Yoksa hata çıkar...

Program size, makine dili programının başlangıç ve bitiş adreslerini soracaktır. Önceden de söylediğimiz gibi, başlangıç adresini bilmeniz gerekiyor. Bitiş adresini bulmayı da öğrendiniz. Program oluşturduğu her satırı ekrana yazacaktır. En sonunda da 500'den sonra gelen satırları silip, DATA satırlarından oluşan programı tek başına bırakacaktır. Artık bu programı, normal bir BASIC programı olduğu için, SAVE ve LOAD ile kullanabilirsiniz. RUN dediğinizde ise, program DATA satırlarını sırayla hafızaya yerleştirecek, o sırada okuduğu satır numarasını gösterecek ve bu arada da her satırı kontrol edecektir. Sırası gelmişken belirtelim, programımız DATA satırlarını oluştururken, her DATA satırına 20 tane hexadecimal (yani onaltılık sistemde) sayı ve bir de kontrol toplamı yazıyor. Böylece hem hafızadan tasarruf yapılıyor, hem de güvenilirlik sağlanıyor.

Dilerseniz, biraz programı inceleyelim. Program o meşhur klavye tamponu (keyboard buffer) ile çalışıyor.

0-10: Oluşacak olan DATA satırlarını hafızaya yerleştirmeye yarayan baş kısım.

500-503: Başlangıç ve bitiş adresleri sorulup, kullanılmayan hafıza bölgelerine yerleştiriliyor. Bunu yapmamızın bir nedeni var. Programda kullanılan yöntem, programa devamlı olarak yeni satırlar eklemekte. Ancak, her yeni satır eklendiğinde, işletim sistemi, o ana kadar tanımlanmış değişkenlerin hepsini otomatik olarak siler. Eğer kullanacağımız değişkenleri hafızada depolamazsak, programımızı yürütemeyiz.

504: D ile kaç satır oluşturduğumuzu sayıyoruz. Ayrıca verilen adreslerin uygunluğu kontrol ediliyor.

505-507: Klavye tamponuna bir HOME ve iki RETURN tuşu yerleştirilip, ekranda iki numaralı satır olarak yeralması gereken FOR...NEXT satırı yazılıyor ve program bitiriliyor. Ancak tampondaki üç tuş otomatik olarak devreye girip, ikinci satırı programa ekliyor ve programı 508 numaralı satırdan itibaren yeniden çalıştırıyor.

508-509: Başlangıç ve bitiş adresleri hafızadan geri okunuyor.

511-512: Onaltılık sistemdeki sayı-

ları oluşturmak için gereken değişkenler hazırlanıyor.

514-521: Sırası gelen hafıza bölgesi DATA satırı haline çevriliyor.

522-525: Hazırlanan DATA satırı aynı yöntemle programın içine bağlanıp, program ya 508'den yeniden başlatılıyor, ya da 526'ya gönderiliyor.

526-529: 500'den sonra gelen satırlar gene klavye tamponu kullanılarak siliniyor.

İşte program bu kadar. Sakın yazarken satır numaralarını değiştirmeyin haaa!..

```
PROGRAM ADI : DATA YARATICI

00000 DIM H(75):FOR I=0 TO 9                                    <129>
00001 H(48+I)=I:H(65+I)=I+10:NEXT                               <048>
00003 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,
      1))                                                       <186>
00004 D=H(H)*16+H(L):S=S+D:POKE I,D                             <197>
00005 A=A+1:IF A<20 THEN NEXT :A=-1                             <110>
00006 PRINT "SATIR:"1000+Z;                                     <164>
00007 READ V:Z=Z+1:IF V=S THEN 9                                <137>
00008 PRINT "TOPLAMA HATASI! SATIR:"999+Z
      :STOP                                                     <173>
00009 IF A<0 THEN END                                           <171>
00010 S=0:A=0:PRINT :NEXT :END                                  <073>
00500 INPUT "BASLANGIC";A                                       <141>
00501 H=INT(A/256):L=A-H*256:POKE 640,H:P
      OKE 641,L                                                 <138>
00502 INPUT "BITIS";E                                           <210>
00503 H=INT(E/256):L=E-H*256:POKE 642,H:P
      OKE 643,L                                                 <155>
00504 POKE 2,1:IF A>E THEN 500                                  <127>
00505 POKE 198,3:POKE 631,19:POKE 632,13:
      POKE 633,13                                               <088>
00506 PRINT "<CLR>2FORI="A"TO"E":READA$":
      PRINT "RUN 508"                                           <195>
00507 END                                                       <255>
00508 A=PEEK(640)*256+PEEK(641)                                 <216>
00509 E=PEEK(642)*256+PEEK(643)                                 <251>
00510 :                                                         <232>
00511 DIM H$(20):FOR I=0 TO 9:H$(I)=CHR$(
      I+48)                                                     <187>
00512 H$(I+10)=CHR$(I+65):NEXT :Z=3                             <227>
00513 :                                                         <237>
00514 FOR I=1 TO 20:D=PEEK(A):S=S+D                             <190>
00515 H=INT(D/16):L=D-16*H                                      <167>
00516 A$=A$+H$(H)+H$(L)+","                                     <068>
00517 IF A=E THEN Z=2:D=PEEK(2):GOTO 519                        <150>
00518 A=A+1:NEXT :D=PEEK(2):POKE 2,D+1                          <218>
00519 A$=STR$(999+D)+"DATA"+A$                                  <124>
00520 A$=A$+STR$(S)                                             <141>
00521 H=INT(A/256):L=A-H*256:POKE 640,H:P
      OKE 641,L                                                 <160>
00522 POKE 631,19:POKE 632,13:POKE 633,13                       <022>
00523 PRINT CHR$(147)A$                                         <062>
00524 PRINT "RUN508":IF Z=2 THEN PRINT "<
      UP>RUN526":Z=3                                            <236>
00525 POKE 198,Z:END                                            <005>
00526 PRINT "<CLR>";:FOR A=500 TO 507:PRI
      NT A:NEXT :FOR A=0 TO 9:POKE 631+A,
      13:NEXT :POKE 631,19:POKE 198,10:PR
      INT "RU527":END                                           <092>
00527 PRINT "<CLR>";:FOR A=508 TO 515:PRI
      NT A:NEXT :FOR A=0 TO 9:POKE 631+A,
      13:NEXT :POKE 631,19:POKE 198,10:PR
      INT "RU528":END                                           <093>
00528 PRINT "<CLR>";:FOR A=516 TO 523:PRI
      NT A:NEXT :FOR A=0 TO 9:POKE 631+A,
      13:NEXT :POKE 631,19:POKE 198,10:PR
      INT "RU529":END                                           <044>
00529 PRINT "<CLR>";:FOR A=524 TO 530:PRI
      NT A:NEXT :FOR A=0 TO 9:POKE 631+A,
      13:NEXT :POKE 631,19:POKE 198,9                           <108>
```

C 128 Püf Noktalarının Programları

PROGRAM 1

```
100 ZN=5100
110 MA=7
120 DIM D(MA),W$(MA)
130 D$="........"
140 REM 80-KOLON KAREKTER TANIMLAYICI
150 FOR Y=0 TO 7:D(Y)=0:NEXT Y
160 PRINT"■■■80-KOLON KAREKTER TANIMLAYICI ■"
170 MF=1
180 PRINT
190 PRINT "DEGISTIRILECEK KAREKTERIN TUSUNA BASIN";:GET KEY C$
200 PRINT C$:PRINT
210 AC=ASC(C$)
220 IF AC AND 128 THEN CO=AC AND 127:GOTO 260
230 IF NOT AC AND 64 THEN CO=AC :GOTO 260
240 IF AC AND 32 THEN CO=AC AND 95 :GOTO 260
250 CO=AC AND 63
260 AD=53248+CO*8
270 C$=""
280 :
290 PRINT "■■■ ORIJINAL■          ■■■ KULLANICI■"
300 BANK 14
310 FOR X=0 TO MA
320 CZ=PEEK(AD+X):IF X>7 THEN CZ=0
330 FOR Y=7 TO 0 STEP -1
340 IF CZ>=2↑Y THEN C$=C$+"■■■":CZ=CZ-2↑Y:ELSE C$=C$+"."
```

```
350 NEXT Y
360 PRINT "■";C$;X;TAB(16);"■";:PRINT USING "##";X;
370 PRINT TAB(19);D$
380 C$=""
390 NEXT X
400 PRINT "■■          ■          ■■          ■"
410 BANK 15
420 :
430 REM YAZMA-YORDAMI
440 PRINT"■"
450 OPEN1,0
460 FOR Y=0 TO MA
470 CHAR,16,6+Y
480 PRINT "->";
490 WINDOW 19,6+Y,27,6+Y,MF
500 POKE 244,1:INPUT#1,W$(Y)
510 W$(Y)=LEFT$(W$(Y),8)
520 WINDOW 0,0,39,24:REM WINDOW 0,0,79,24 80-KOLON ICIN
530 FOR X=1 TO 8
540 IF MID$(W$(Y),X,1)="*" THEN D(Y)=D(Y)+2↑(8-X)
550 NEXT X
560 NEXT Y
565 CLOSE1
570 CHAR ,0,22:PRINT"DUZELTME (E/H) ?";:GET KEY A$
580 IF A$="E" THEN MF=0:FOR Y=0 TO 7:D(Y)=0:NEXT Y:GOTO430
590 PRINT:PRINT"KAREKTER KULLANILSINMI (E/H) ?";:GET KEY A$
600 IF A$="H" THEN 150
1000 REM KAREKTER KULLANIMI
1010 PRINT"■■■";
1020 ZN=ZN+10
1030 PRINT ZN;"DATA ";CO;",";
1040 FOR X=0 TO MA:PRINT D(X);"■,";:NEXT X
1050 PRINT CHR$(20)
1060 PRINT ZN+10;"DATA -1"
1070 PRINT :PRINT "GOTO 30"
1080 PRINT
1090 FOR Y=842 TO 845:POKE Y,13:NEXT Y
1100 POKE 208,4
1110 END
5000 REM KAREKTER TANIMLAMA BASIC YUKLEYICISI
5005 FOR X=6144 TO 6182
5010 READ A
5015 CS=CS+A
5020 POKE X,A
5025 NEXT X
5030 IF CS<>3411 THEN PRINT CHR$(7):LIST 5035-5050:END
5035 DATA  72,138, 72,152, 72,169,  2,141, 40, 10,162, 18
5040 DATA 104, 32, 27, 24,232,104, 32, 27, 24,162, 31
```

```
5045 DATA 104, 76, 27, 24,142,  0,214, 44,  0,214, 16,251
5046 DATA 141,  1,214, 96
5050 READ A:IF A=-1 THEN END
5055 FOR Y=0 TO 7
5060 READ W
5065 AD=8192+16*A+Y
5070 HI=INT(AD/256):LO=AD-(256*HI)
5075 SYS DEC("1800"),W,LO,HI
5080 NEXT Y
5085 GOTO 5050
5110 DATA 1,0,0,0,0,0,0,0,0
5120 DATA 1,0,0,0,0,0,0,0,0
5130 DATA -1

READY.
```

PROGRAM 2

```
10 A=DEC("D600"):B=A+1
20 FOR X=0 TO 8
30 POKE A,22:POKE B,PEEK(B) AND NOT 7 OR X
40 FOR T=1 TO 100:NEXT T
50 NEXT X
60 FOR X=0 TO 8
70 POKE A,23 :POKE B,X
71 FOR T=1 TO 100 :NEXT T
80 NEXT X
90 END

READY.
```

PROGRAM 3

```
10 A=DEC("D600"):B=A+1
20 FOR X=0 TO 8
30 POKE A,22:POKE B,PEEK(B) AND NOT 7 OR X
40 POKE A,23:POKE B,X
50 FOR T=1 TO 200:NEXT T
60 NEXT X
70 END
```

C 64 ile Grafik (8)'in Programları

BANKSWICH

```
10 BANK=3:EKRAN=3:GRAFIK=1:KARAKTER=3
20 POKE56576,PEEK(56576)AND252OR(3-BANK)
30 POKE53248+24,PEEK(53248+24)AND15OREKRAN*16
40 POKE53248+24,PEEK(53248+24)AND241ORKARAKTER*2
50 POKE648,(BANK*16*1024+EKRAN*1024)/256
55 VIDEO=BANK*16*1024+EKRAN*1024:FORI=0TO999:POKEVIDEO+I,PEEK(1024+I):NEXT
60 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251:B1=53248:B2=57343
70 FORI=B1TOB2:POKEI,PEEK(I):NEXT
80 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
READY.


 10 REM
 20 REM ****************************
 30 REM * KARAKTER SETI EDITORU *
 40 REM ****************************
 50 REM *        (C)  1986         *
 60 REM *      TUNA ERTEMALP       *
 70 REM *    AYHAN KALAYLIOGLU     *
 80 REM ****************************
 90 REM
100 POKE55,0:POKE56,32:CLR:POKE650,128
110 POKE53280,14:POKE53281,6:POKE646,14
120 DIMC(8,8),D(8,8),B(7):KD=0:X=2:Y=7:Z=8:W=1
130 Z1=1024:Z4=40:Z3=32:Z6=6:Z5=15:Z0=16
140 FORI=.TOY:B(I)=X↑(Y-I):NEXT
150 INPUT"■ BASLANGIC ADRESINI GIRIN";BS:KD=0
160 IFBS>=4*4096ORBS/2048<>INT(BS/2048)ORBS<2*4096THEN150
170 GOTO430
180 PRINT"■";:FORI=.TOY:FORJ=.TOY:IFC(I,J)THENPRINT"*";:GOTO200
190 PRINT" ";
```

```
200 NEXT:PRINT"■ ":NEXT:PRINT"■          ■"RIGHT$("000"+MID$(STR$(KD),2),3)
210 PRINT"■"TAB(15)"    [F1] HAFIZADAN AL"
220 PRINTTAB(15)"     [F2] HAFIZAYA KOY"
230 PRINTTAB(15)"[X] [F3] X-SIMETRI"
240 PRINTTAB(15)"[Y] [F4] Y-SIMETRI"
250 PRINTTAB(15)"[+] [F5] + YONDE DONDUR"
260 PRINTTAB(15)"[-] [F6] - YONDE DONDUR"
270 PRINTTAB(15)"     [F7] EDIT"
280 PRINTTAB(15)"     [F8] REVERSE"
290 PRINTTAB(15)"     [C ] KARAKTER SETI"
300 PRINTTAB(15)"[.] [> ] BIR SONRAKI"
310 PRINTTAB(15)"[,] [< ] BIR ONCEKI"
320 PRINTTAB(15)"     [A ] YENI ADRES"
330 PRINTTAB(15)"     [O ] ORIJINAL SET"
340 PRINTTAB(15)"     [L ] LOAD SET"
350 PRINTTAB(15)"     [S ] SAVE SET"
360 PRINTTAB(15)"     [Q ] PROGRAMI BITIR"
370 PRINTTAB(15)"    CRSR KAYDIRMA"
380 POKE198,0:WAIT198,1:GETA$
390 IFA$<>"■"ANDA$<>"■"ANDA$<>"■"ANDA$<>"Q"THENPRINT"LUTFEN BEKLEYIN..  "
400 IFA$="A"THEN150
410 IFA$<>"■"THEN450
420 INPUT"KARAKTER KODU ";KD
430 FORI=.TOY:FORJ=.TOY:C(I,J)=-((PEEK(BS+Z*KD+I)AND(B(J))))>.):NEXTJ,I
440 GOTO180
450 IFA$<>"■"THEN490
460 INPUT"KARAKTER KODU ";KD
470 FORI=.TOY:B=.:FORJ=.TOY:B=B+C(I,J)*B(J):NEXTJ:POKEBS+Z*KD+I,B:NEXTI
480 GOTO430
490 IFA$<>"■"ANDA$<>"X"THEN530
500 FORI=.TOY:FORJ=.TOY:D(Y-I,J)=C(I,J):NEXTJ,I
510 FORI=.TOY:FORJ=.TOY:C(I,J)=D(I,J):NEXTJ,I
520 GOTO180
530 IFA$<>"■"ANDA$<>"Y"THEN570
540 FORI=.TOY:FORJ=.TOY:D(I,Y-J)=C(I,J):NEXTJ,I
550 FORI=.TOY:FORJ=.TOY:C(I,J)=D(I,J):NEXTJ,I
560 GOTO180
570 IFA$<>"■"ANDA$<>"+"THEN610
580 FORI=.TOY:FORJ=.TOY:D(Y-J,I)=C(I,J):NEXTJ,I
590 FORI=.TOY:FORJ=.TOY:C(I,J)=D(I,J):NEXTJ,I
600 GOTO180
610 IFA$<>"■"ANDA$<>"-"THEN650
620 FORI=.TOY:FORJ=.TOY:D(J,Y-I)=C(I,J):NEXTJ,I
630 FORI=.TOY:FORJ=.TOY:C(I,J)=D(I,J):NEXTJ,I
640 GOTO180
650 IFA$<>"■"THEN700
660 POKE631,20:POKE632,20:POKE633,19:POKE198,3:INPUTA$
670 FORI=.TOY:FORJ=.TOY:C(I,J)=W:IFPEEK(Z1+Z4*I+J)=Z3THENC(I,J)=.
680 NEXTJ,I
690 GOTO180
700 IFA$<>"■"THEN730
710 FORI=.TOY:FORJ=.TOY:C(I,J)=W-C(I,J):NEXTJ,I
720 GOTO180
730 IFA$<>"■"THEN770
740 FORI=YTOWSTEP-W:FORJ=.TOY:C(I,J)=C(I-W,J):NEXTJ,I
750 FORJ=.TOY:C(.,J)=.:NEXT
760 GOTO180
770 IFA$<>"□"THEN810
780 FORI=.TOZ6:FORJ=.TOY:C(I,J)=C(I+W,J):NEXTJ,I
790 FORJ=.TOY:C(Y,J)=.:NEXT
```

```
800 GOTO180
810 IFA$<>"■"THEN850
820 FORI=.TOY:FORJ=YTOWSTEP-W:C(I,J)=C(I,J-W):NEXTJ,I
830 FORI=.TOY:C(I,.)=.:NEXT
840 GOTO180
850 IFA$<>"■■"THEN890
860 FORI=.TOY:FORJ=.TOZ6:C(I,J)=C(I,J+W):NEXTJ,I
870 FORI=.TOY:C(I,Y)=.:NEXT
880 GOTO180
890 IFA$<>"C"THEN950
900 POKE53281,5:PRINT"■";:POKE53281,0:POKE53280,0:TP=PEEK(53272)
910 POKE53272,BS/2048*2+16
920 FORI=.TOZ5:FORJ=.TOZ5:POKEZ1+Z4*I+J*X,Z0*I+J:NEXTJ,I
930 POKE198,0:WAIT198,1:POKE53272,TP:POKE53281,6:POKE53280,14
940 GOTO180
950 IFA$<>">"ANDA$<>"."THEN980
960 KD=KD+W:IFKD=512THENKD=.
970 GOTO430
980 IFA$<>"<"ANDA$<>","THEN1010
990 KD=KD-W:IFKD=-WTHENKD=511
1000 GOTO430
1010 IFA$<>"Q"THEN1060
1020 PRINT"EMIN MISINIZ ? (E/H)"
1030 POKE198,0:WAIT198,1:GETA$:IFA$<>"E"ANDA$<>"H"THEN1030
1040 IFA$="E"THENPRINT"■";:END
1050 GOTO180
1060 IFA$<>"O"THEN1150
1070 PRINT"□EMIN MISINIZ ? (E/H)"
1080 POKE198,0:WAIT198,1:GETA$:IFA$<>"E"ANDA$<>"H"THEN1080
1090 IFA$="H"THEN180
1100 PRINT"□LUTFEN BEKLEYINIZ..."
1110 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251:ZX=53248:PRINT
1120 FORI=0TO4095:POKEBS+I,PEEK(ZX+I):NEXT
1130 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
1140 GOTO430
1150 IFA$<>"L"THEN1190
1160 A$="■■■"
1170 INPUT"□DOSYA ADINI GIRIN ";A$:YX=PEEK(186):IFA$="■■■"THEN1170
1180 LOADA$,YX,1
1190 IFA$<>"S"THEN1260
1200 A$="■■■"
1210 INPUT"□DOSYA ADINI GIRIN ";A$:YX=PEEK(186):IFA$="■■■"THEN1210
1220 H1=INT(BS/256):L1=BS-H1*256
1230 H2=INT((BS+4096)/256):L2=(BS+4096)-H2*256
1240 SYS(57812)A$,YX:POKE193,L1:POKE194,H1:POKE174,L2:POKE175,H2:SYS62957
1250 GOTO180
1260 GOTO180
READY.
```